



Dialogic® Vision™ VX

Integrated Media Platform

User's Manual

Copyright and legal notice

Copyright © 2006-2009 Dialogic Corporation. All Rights Reserved. You may not reproduce this document in whole or in part without permission in writing from Dialogic Corporation at the address provided below.

All contents of this document are furnished for informational use only and are subject to change without notice and do not represent a commitment on the part of Dialogic Corporation or its subsidiaries ("Dialogic").

Reasonable effort is made to ensure the accuracy of the information contained in the document. However, Dialogic does not warrant the accuracy of this information and cannot accept responsibility for errors, inaccuracies or omissions that may be contained in this document.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH DIALOGIC® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN A SIGNED AGREEMENT BETWEEN YOU AND DIALOGIC, DIALOGIC ASSUMES NO LIABILITY WHATSOEVER, AND DIALOGIC DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF DIALOGIC PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT OF A THIRD PARTY.

Dialogic products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Due to differing national regulations and approval requirements, certain Dialogic products may be suitable for use only in specific countries, and thus may not function properly in other countries. You are responsible for ensuring that your use of such products occurs only in the countries where such use is suitable. For information on specific products, contact Dialogic Corporation at the address indicated below or on the web at www.dialogic.com.

It is possible that the use or implementation of any one of the concepts, applications, or ideas described in this document, in marketing collateral produced by or on web pages maintained by Dialogic may infringe one or more patents or other intellectual property rights owned by third parties. Dialogic does not provide any intellectual property licenses with the sale of Dialogic products other than a license to use such product in accordance with intellectual property owned or validly licensed by Dialogic and no such licenses are provided except pursuant to a signed agreement with Dialogic. More detailed information about such intellectual property is available from Dialogic's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. **Dialogic encourages all users of its products to procure all necessary intellectual property licenses required to implement any concepts or applications and does not condone or encourage any intellectual property infringement and disclaims any responsibility related thereto. These intellectual property licenses may differ from country to country and it is the responsibility of those who develop the concepts or applications to be aware of and comply with different national license requirements.**

Any use case(s) shown and/or described herein represent one or more examples of the various ways, scenarios or environments in which Dialogic® products can be used. Such use case(s) are non-limiting and do not represent recommendations of Dialogic as to whether or how to use Dialogic products.

Dialogic, Dialogic Pro, Brooktrout, Diva, Cantata, SnowShore, Eicon, Eicon Networks, NMS Communications, NMS (stylized), Eiconcard, SIPcontrol, Diva ISDN, TruFax, Exnet, EXS, SwitchKit, N20, Making Innovation Thrive, Connecting to Growth, Video is the New Voice, Fusion, Vision, PacketMedia, NaturalAccess, NaturalCallControl, NaturalConference, NaturalFax and Shiva, among others as well as related logos, are either registered trademarks or trademarks of Dialogic Corporation or its subsidiaries. Dialogic's trademarks may be used publicly only with permission from Dialogic. Such permission may only be granted by Dialogic's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. Any authorized use of Dialogic's trademarks will be subject to full respect of the trademark guidelines published by Dialogic from time to time and any use of Dialogic's trademarks requires proper acknowledgement.

The names of actual companies and products mentioned herein are the trademarks of their respective owners.

This document discusses one or more open source products, systems and/or releases. Dialogic is not responsible for your decision to use open source in connection with Dialogic products (including without limitation those referred to herein), nor is Dialogic responsible for any present or future effects such usage might have, including without limitation effects on your products, your business, or your intellectual property rights.

Revision history

P/N 64-0400-01

Revision	Release date	Notes
A	June 2009	BK, Dialogic® Vision™ VX Integrated Media Platform 4.1
Last modified: June 19, 2009		

Table Of Contents

Chapter 1: Introduction	7
Chapter 2: Overview of the VX Integrated Media Platform	9
VX Integrated Media Platform overview	9
Software components.....	9
Signaling protocols and models.....	10
Media capabilities	10
Interfaces.....	11
Fast call setup.....	12
ISDN models	12
ISDN audio model	12
ISDN video model.....	13
ISUP models.....	14
Basic ISUP audio model	14
Basic ISUP video model	14
ISUP scalable deployment model	15
ISUP redundant deployment model	16
IP models.....	17
IP-only audio model.....	17
IP-only video model	17
Standards	19
Document conventions.....	20
Related documentation	20
Chapter 3: Configuring the VX Integrated Media Platform	21
Overview of configuring the VX Integrated Media Platform	21
Gathering information	22
Network configuration information	22
ISDN configuration information.....	22
Trunk configuration information.....	23
Signaling server configuration information.....	24
ISUP configuration information	25
Logging into the platform for the first time.....	28
Configuring the platform to use a static IP address.....	28
Obtaining an IP address through DHCP	29
Accessing the Vision Console.....	30
Creating or revising a configuration.....	31
Additional configuration tasks.....	31
Backing up a configuration.....	33
Restoring a configuration.....	34
Accessing the platform using a secure shell.....	35
Chapter 4: Vision™ Console parameters.....	37
Configuration menu parameters	37
Overview.....	37
Host IP information.....	38
Resource configuration	39
SIP/RTP parameters.....	40
Trunks	41
PSTN	43

Signaling Server	45
Options	48
Port upgrade	48
Import/Export configuration	48
Operations menu parameters	49
Services	49
Maintenance	49
Provisioning menu parameters	51
Call routing table	51
CCXML application configuration	51
VoiceXML application configuration	51
Monitoring menu parameters	52
Trunks	53
CCXML statistics	53
Call Server status	54
VoiceXML Interpreter view	55
VoiceXML Interpreter status	55
SSML Processor view	56
SSML Processor status	57
Signaling Monitor	58
Chapter 5: Managing the VX Integrated Media Platform	59
Working with VX Integrated Media Platform services	59
Viewing platform information	60
Managing VoiceXML applications	61
Adding a VoiceXML application to the Vision Console	62
Removing a VoiceXML application from the Vision Console	63
Modifying a VoiceXML application definition	63
Pattern matching syntax	64
Managing CCXML applications	65
Default CCXML application	65
Adding a CCXML application definition to the Vision Console	65
Removing a CCXML application definition from the Vision Console	67
Modifying a CCXML application definition	67
CCXML application definition pattern matching syntax	68
Chapter 6: Developing VoiceXML applications	69
Overview of developing VoiceXML applications	69
Managing media	69
Developing an audio application	70
Developing a video application	71
Playing a video message	71
Recording and posting a video message	71
MPEG-4 limitation for 3G-324M handsets	72
Using the record utterance functionality	73
Chapter 7: VX Integrated Media Platform logging	75
Overview of VX Integrated Media Platform logging	75
Call Server system logs	76
Logging levels	76
Logging defaults	76
Changing the logging level	77
Changing other logging defaults	77

Log file format	77
VoiceXML Interpreter system logs	79
Logging levels	79
Logging defaults	80
Changing the logging level	80
Changing other logging defaults	80
Log file format	80
SSML Processor system logs	82
Logging levels	82
Logging defaults	82
Changing the logging level	83
Changing other logging defaults	83
Log file format	83
VoiceXML application logs	85
Log file format	85
Log file format	85
SSML application logs	88
Logging and performance	89
Chapter 8: CallPlacer interface	91
Overview of the CallPlacer interface	91
Request XML	92
Response XML	93
Callback XML	94
CallPlacer XML document type definition (DTD)	96
Chapter 9: Authorization and Usage Indication interface	97
Overview of the Authorization and Usage Indication interface	97
Message format	98
HTTP operations	98
Authorization exchange	99
AuthorizationRequest	99
AuthorizationResponse	100
Usage exchange	102
UsageIndication request message	102
UsageConfirmation message	104
Call detail records	105
CDR entry format	105
Chapter 10: SIP interface	109
Overview of the SIP interface	109
Interoperability	110
Inbound calls	111
Variable mappings	112
Specifying the initial VoiceXML URI to invoke	112
Session data update	113
CCXML parameters	113
Proxy server	114
Location server	114
Load balancing and failover	114
Outbound calls	115
VoiceXML exit information	116
Returned data encoded as SIP-VXML	116

Returning data in XML format.....	116
BYE reason header	118
Call transfer	119
Bridge transfer.....	119
Blind transfer.....	120
Consultation transfer.....	121
Chapter 11: Fine tuning the VX Integrated Media Platform configuration	123
Overview of fine tuning the configuration.....	123
Fine tuning the media board configuration	123
oamsys.cfg file.....	123
Media board configuration files	124
Fine tuning the telecom configuration.....	127
Settings for all models except IP-only audio and IP-only video	127
Setting for ISDN models.....	130
Settings for the ISUP models that use Vision™ Signaling Servers.....	131
Settings for ISUP models that use native ISUP signaling.....	132
Fine tuning the H.100 clocking configuration	133
Default H.100 clocking configuration	133
H.100 clock manager configuration file	134
Changing the default H.100 clocking configuration.....	136
Chapter 12: Glossary	137

1 Introduction

The *Dialogic® Vision™ VX Integrated Media Platform User's Manual* describes how to configure and develop applications for the Dialogic® Vision™ VX Integrated Media Platform. This manual assumes that you are familiar with the VoiceXML language and coding web applications.

For information on installing the VX Integrated Media Platform hardware, see *Installing the Dialogic® Vision™ Server TIGI2U*.

Note: The products to which this document pertains are part of the NMS Communications Platforms business that was sold by NMS Communications Corporation ("NMS") to Dialogic Corporation ("Dialogic") on December 8, 2008. Accordingly, certain terminology relating to the products has been changed. Below is a table indicating both terminology that was formerly associated with the products, as well as the new terminology by which the products are now known.

Former terminology	Current terminology
Vision CX Gateway	Dialogic® Vision™ CX Video Gateway
Vision VoiceXML Server	Dialogic® Vision™ VX Integrated Media Platform

The term Vision™ Server is used in this manual to refer to both the CX Video Gateway and the VX Integrated Media Platform collectively.

2

Overview of the VX Integrated Media Platform

VX Integrated Media Platform overview

The VX Integrated Media Platform is a VoiceXML media server that enables users to deploy video and speech-enabled applications in IP, ISDN, and ISUP networks. It provides the following functionality:

- Supports industry-standard VoiceXML 2.1 and the use of CCXML to control inbound and outbound dialing, call transfers, and conferencing.
- Supports the HTTP streaming of media files with caching on the server.
- Optionally provides in-band DTMF support.
- Optionally connects incoming faxes to a third-party T.38 server.
- Optionally provides fast call setup techniques to speed up 3G-324M call setup time.
- Produces detailed event logs with multiple information levels.
- Provides a web-based console, called the Vision™ Console, to configure and manage the VX Integrated Media Platform.
- Provides an SNMP interface for monitoring application usage and server health.

The VX Integrated Media Platform is described in more detail in the following topics:

- Software components
- Signaling protocols and models
- Media capabilities
- Interfaces
- Fast call setup

Software components

The VX Integrated Media Platform contains the Call Server and the VoiceXML Subsystem.

The Call Server manages call control and transaction capabilities for all VX Integrated Media Platform models except IP-only audio.

The VoiceXML Subsystem includes the following software components:

Component	Description
VoiceXML Interpreter	Interprets VoiceXML dialogs. At startup, the VoiceXML Interpreter loads a media resource object named MIOSIP. This object provides the Media Resource Function with the ability to process media.
SSML Processor	Processes SSML requests for multimedia and text-to-speech. At startup, the SSML Processor loads an audio driver TCP object named ADTCP. This object serves as a link between the SSML Processor and the VoiceXML Interpreter.
Media Resource Function	Provides media processing including record, playback, and interfaces to speech recognition resources.

Signaling protocols and models

The VX Integrated Media Platform can implement the following signaling protocols:

- ISDN, which is available with the ISDN audio and ISDN video models.
- ISUP, which is available with the ISUP audio and ISUP video models. ISUP models can be configured for scalability and redundant capability.

All VX Integrated Media Platform models use SIP/RTP signaling to interface with the IP network. For more information, see *ISDN models* on page 12, *ISUP models* on page 14, and *IP models* on page 17.

Media capabilities

All VX Integrated Media Platform models support the following audio media formats:

Format	Encoding
Raw (headerless)	8 kHz 8-bit mu-law and A-law (G.711)
WAV (RIFF header)	8 kHz 8-bit mu-law and A-law, and 8 kHz 16-bit linear
Sun Audio (AU)	8 kHz 8-bit mu-law and A-law, and 8 kHz 16-bit linear
SPHERE (NIST header)	8 kHz 8-bit mu-law, and 8 kHz 16-bit linear

The ISDN video, ISUP video, and IP-only video models also support the video/3gpp MIME type.

The media capabilities in the platform support the following codecs:

Codec type	Description
Audio	• AMR (3GPP TS 26.090, 26.101, and 26.073, Version 5.3.0, 2004) • G.711 A-law and mu-law • G.723.1 • G.726 • G.729 A
Video	• H.263, H.263+ • H.264 • MPEG-4

Interfaces

The VX Integrated Media Platform includes the following programmatic and management interfaces:

Interface	Description
VoiceXML	W3C standardized interface for creating user dialogs.
Call Control Extensible Markup Language (CCXML)	W3 interface that enables the VX Integrated Media Platform to execute applications written in the W3C Working Draft of CCXML dated 29th June 2005 (http://www.w3.org/TR/2005/WD-ccxml-20050629). You can use CCXML to write applications that provide call control for the duration of a phone call, including call setup, monitoring, and teardown. You can also use CCXML applications to provide call routing and conferencing functionality. For information, see <i>Managing CCXML applications</i> on page 65, the <i>Dialogic® Vision™ CCXML Developer's Manual</i>, and http://www.w3.org/TR/2005/WD-ccxml-20050629.
CallPlacer interface	XML-over-HTTP mechanism for initiating outbound sessions or calls for VoiceXML applications. For information, see <i>Overview of the CallPlacer interface</i> on page 91.
Authorization and Usage Indication interface	XML-over-HTTP mechanism for access control over session establishment and gathering information for call detail reports. For information, see <i>Overview of the Authorization and Usage Indication interface</i> on page 97.
SIP interface	Signaling interface presenting calls or a method of establishing sessions for an application server. For information, see <i>Overview of the SIP interface</i> on page 109.
SNMP	Agent that provides enterprise MIBs to monitor server performance, view statistics, monitor a RAID server, and receive alarms. The master agent supports SNMPv1, SNMPv2c, and SNMPv3. For information, see the <i>Dialogic® Vision™ SNMP Reference Manual</i>.

Fast call setup

The VX Integrated Media Platform supports the following techniques for speeding up 3G-324M call setup time:

- Packed H.245 messages, which group independent H.245 messages together into a single NSRP command frame. This reduces the number of message round-trips, and thus reduces call setup time.
- Windowed Simple Retransmission protocol (WNSRP), an H.245 transport improvement technique that is standardized in ITU-T Recommendation H.324 and accepted into the 3G-324M standard by 3GPP.
- Media oriented negotiation acceleration (MONA), which unites the technologies for H.324 call setup acceleration under a common framework. MONA was approved by the ITU-T in August 2006, and is recommended in 3GPP Release 7 in TR 26.911.

Note: You must obtain the appropriate license to use one of these techniques. For information, see the *readme* file for this release.

The VX Integrated Media Platform does not support:

- The signaling preconfigured channel (SPC) MONA technique.
- Preconfigured channel media frames encapsulated in MONA signaling preference messages.

ISDN models

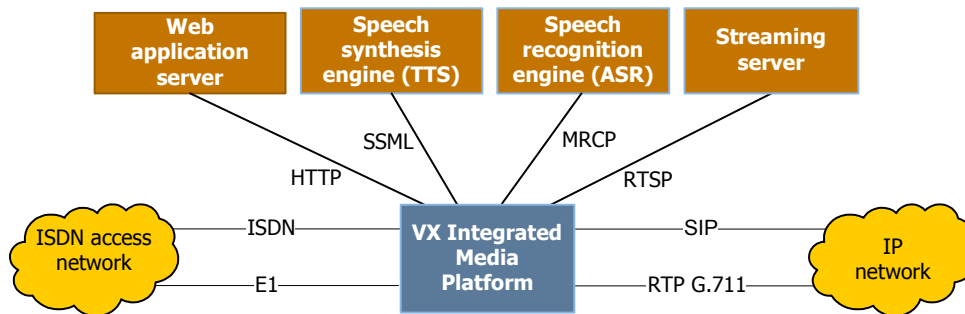
The VX Integrated Media Platform is available with an ISDN audio model and an ISDN video model.

ISDN audio model

In the ISDN audio model, the platform:

- Provides ISDN signaling.
- Provides SIP signaling.
- Supports the Call Control Extensible Markup Language (CCXML) for call control.
- Executes VoiceXML applications and their associated media processing functions.

The following illustration shows the ISDN audio model:



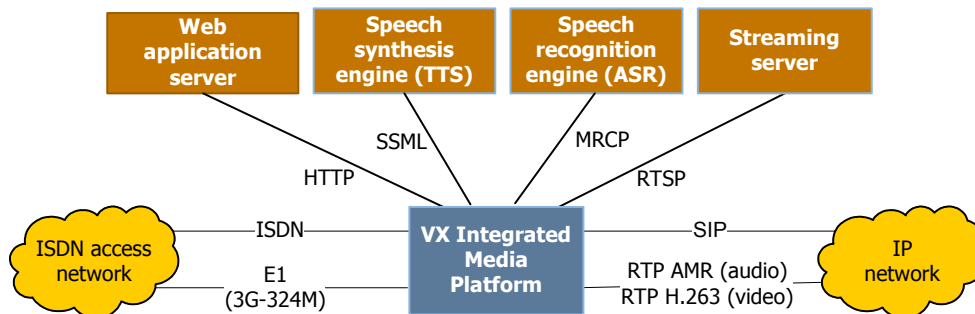
ISDN video model

In the ISDN video model, the platform:

- Provides ISDN signaling with 3G-324M.
- Provides SIP signaling.
- Provides the option to transcode between AMR and G.711.
- Supports the Call Control Extensible Markup Language (CCXML) for call control.
- Executes VoiceXML applications and their associated media processing functions.

The ISDN video model requires a mobile video device that supports 3G-324M.

The following illustration shows the ISDN video model:



ISUP models

The VX Integrated Media Platform is available with an ISUP audio model and an ISUP video model. Each of these models can be configured for scalability and redundant capability.

This topic describes the:

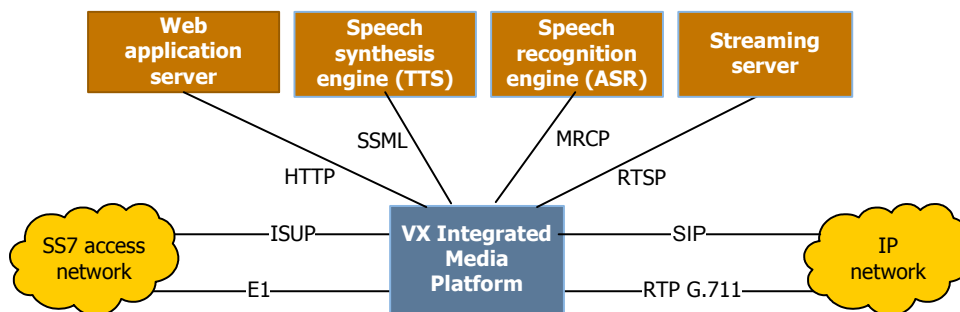
- Basic ISUP audio model
- Basic ISUP video model
- ISUP scalable deployment model
- ISUP redundant deployment model

Basic ISUP audio model

In the basic ISUP audio model, the platform:

- Provides ISUP signaling.
- Provides SIP signaling.
- Supports the Call Control Extensible Markup Language (CCXML) for call control.
- Executes VoiceXML applications and their associated media processing functions.

The following illustration shows the basic ISUP audio model:



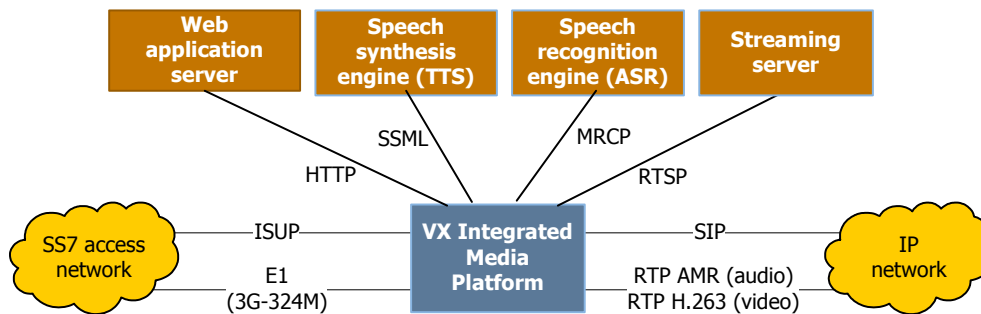
Basic ISUP video model

In the basic ISUP video model, the platform:

- Provides ISUP signaling with 3G-324M.
- Provides SIP signaling.
- Supports the Call Control Extensible Markup Language (CCXML) for call control.
- Executes VoiceXML applications and their associated media processing functions.

The basic ISUP video model requires a mobile video device that supports 3G-324M.

The following illustration shows the basic ISUP video model:

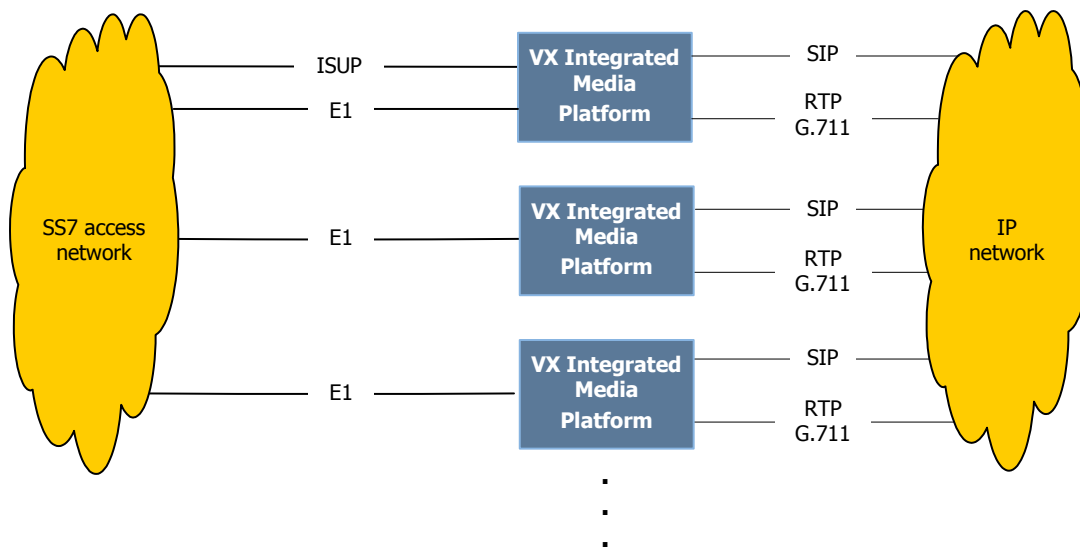


ISUP scalable deployment model

For scalability, multiple VX Integrated Media Platforms can be deployed where the ISUP termination on one server provides the signaling between the SS7 access network and each of the other servers in the system.

In this model, the servers are configured to share the single signaling point code terminated by the platform with ISUP interface. This configuration allows for high density deployments for a single signaling point code.

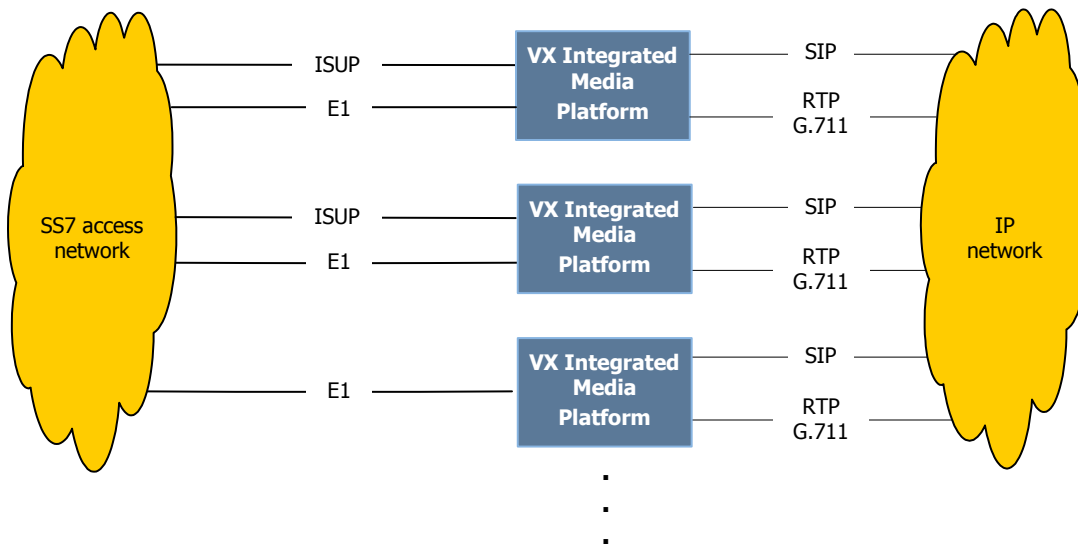
The following illustration shows the ISUP scalable deployment model.



ISUP redundant deployment model

For a redundant and fault-tolerant system, two VX Integrated Media Platforms can be deployed to provide higher availability. The servers share a single signaling point code and provide node-level redundancy.

The following illustration shows an ISUP model with two VX Integrated Media Platforms to support redundancy. The redundant pair of servers with ISUP terminations seamlessly provides signaling services for multiple servers as in the scalable deployment model.



IP models

The VX Integrated Media Platform is available with an IP-only audio model and an IP-only video model.

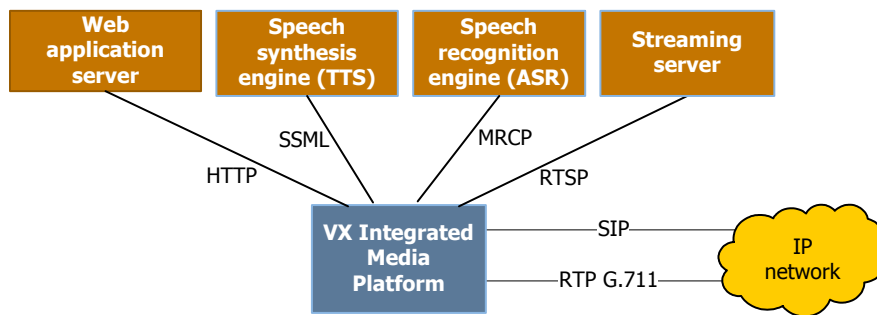
IP-only audio model

In the IP-only audio model, the platform:

- Provides SIP signaling.
- Provides audio media capabilities.
- Supports up to 240 simultaneous audio calls.
- Executes VoiceXML applications and their associated media processing functions.

The web application uses standard VoiceXML operations to process audio. For information, see *Developing an audio application* on page 70 and the *Dialogic® Vision™ VoiceXML Reference Manual*.

The following illustration shows the IP-only audio model:



Note: The IP-only audio model does not use media boards and does not include the Call Server.

IP-only video model

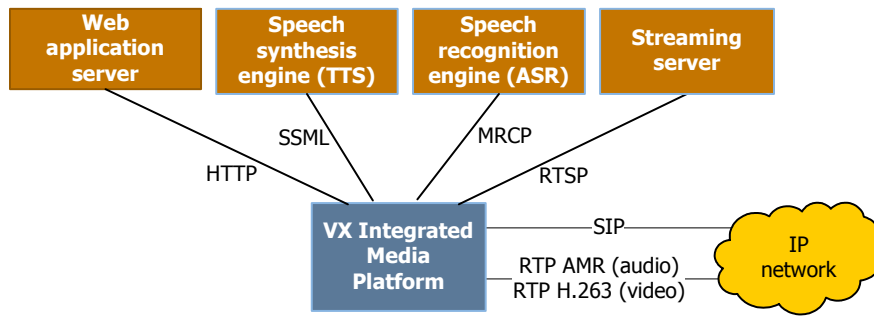
In the IP-only video model, the platform:

- Provides SIP signaling.
- Provides audio capabilities.
- Provides video capabilities.
- Provides the option to transcode between AMR to G.711.
- Supports up to 240 simultaneous video calls, audio calls, or a combination of both.
- Supports the Call Control Extensible Markup Language (CCXML) for call control.
- Executes VoiceXML applications and their associated media processing functions.

The web application uses standard VoiceXML operations to process audio, and extended VoiceXML operations to process video. It can play, record, and post video messages.

For more information, see *Developing an audio application* on page 70, *Developing a video application* on page 71, and the *Dialogic® Vision™ VoiceXML Reference Manual*.

The following illustration shows the IP-only video model:



Standards

The VX Integrated Media Platform complies with and supports the following standards, depending on the model:

Standard	Version	VX Integrated Media Platform model
CCXML	Version 1.0, based upon the W3C Working Draft of CCXML dated 29th June 2005 See http://www.w3.org/TR/2005/WD-ccxml-20050629 .	All models, except IP-only audio
ISUP	ITU White, ITU Blue, ITU 97, ETSI V2, ETSI V3, Q.767, ANSI 88, ANSI 92, ANSI 95, JNTT	ISUP audio and ISUP video
MRCP	"A Media Resource Control Protocol Developed by Cisco, Nuance, and Speechworks", Shanmugham, Monaco, and Eberman, IETF Internet-Draft, draft-shanmugham-mrcp-05, January 2004.	All models
MTP	ITU, ANSI, JNTT, JTTC	ISUP audio and ISUP video
NETANN	"Basic Network Media Services with SIP", IETF RFC 4240 See http://www.ietf.org/rfc/rfc4240.txt .	All models that use a SIP application server
OSP	Open Settlement Protocol ETSI TS 101 321	All models
RFC 2833	"RTP Payload for DTMF Digits, Telephony Tones and Telephony Signal", Schulzrinne and Petrack, IETF RFC 2833, May 2000.	All models
RTSP	"Real Time Streaming Protocol (RTSP)", Schulzrinne, Rao, and Lanphier, IETF RFC 2326, April 1998.	All models
SIP	<i>SIP: Session Initiation Protocol</i> , Rosenberg et al., IETF RFC 3261, June 2002	All models
SIP-VXML	"SIP Interface to VoiceXML Media Services", Burke et al., IETF Internet-Draft, draft-burke-vxml-2, November 2006	All models
SISR	W3C Semantic Interpretation for Speech Recognition, Version 1.0	All models
SRGS	"Speech Recognition Grammar Specification Version 1.0", Hunt et al, W3C Candidate Recommendation, 16th March 2004. See http://www.w3.org/TR/speech-grammar .	All models
SSML	"Speech Synthesis Markup Language Version 1.0", Burnett et al, W3C Recommendation, 7th September 2004. See http://www.w3.org/TR/speech-synthesis .	All models
VoiceXML	"Voice Extensible Markup Language (VoiceXML) Version 2.1", Oshry et al, W3C Candidate Recommendation, 13th June 2005. See http://www.w3.org/TR/voicexml21 .	All models

Document conventions

By default, the VX Integrated Media Platform software is installed in the `/opt/nms/vx` directory. This manual uses the string `vx` to refer to the default installation directory.

Related documentation

The following manuals provide information related to installing and configuring the VX Integrated Media Platform:

Document	Description
<i>Installing the Dialogic® Vision™ TIGI2U</i>	Describes how to install and cable the Vision™ hardware.
<i>Dialogic® Vision™ VoiceXML Administration Manual</i>	Describes how to configure the VoiceXML subsystem. The information in this document supplements the VoiceXML subsystem configuration information in this manual.
<i>Dialogic® Vision™ Call Server Administration Manual</i>	Describes how to configure the Call Server. The information in this document supplements the Call Server configuration information in this manual.
<i>Dialogic® Vision™ Signaling Server Administration Manual</i>	Describes how to configure the Vision™ Signaling Server. The information in this document supplements the Vision™ Signaling Server configuration information in this manual.
<i>Dialogic® Vision™ CCXML Developer's Manual</i>	Describes how to use the CCXML interface to configure and develop CCXML applications for the VX Integrated Media Platform.
<i>Dialogic® Vision™ SNMP Reference Manual</i>	Describes the management information bases (MIBs) and agents that support SNMP on the VX Integrated Media Platform.
<i>Dialogic® Vision™ VoiceXML Reference Manual</i>	An online manual that provides a VoiceXML language dictionary.
<i>CG 6565 Installation and Developer's Manual</i>	Describes how to configure the media boards. The information in the <i>CG 6565 Installation and Developer's Manual</i> supplements the media board configuration information in this manual.

3

Configuring the VX Integrated Media Platform

Overview of configuring the VX Integrated Media Platform

All software is pre-installed and pre-configured on the VX Integrated Media Platform. However, software parameters are set for the manufacturing environment. You must re-configure some of these parameters so that the system operates properly at your site.

Use the Vision™ Console to re-configure the server. Using this tool, you enter field values and the tool automatically modifies the configuration files for your model.

This section describes how to use the Vision™ Console to set up the VX Integrated Media Platform software. It contains the following topics:

- Gathering information
- Logging into the server for the first time
- Accessing the Vision Console
- Creating or revising a configuration
- Backing up a configuration
- Restoring a configuration
- Accessing the platform using a secure shell

Note: After you create the VX Integrated Media Platform configuration, you can fine tune it if necessary. For information, see *Overview of fine tuning the configuration* on page 123.

Gathering information

Before you configure the VX Integrated Media Platform, have the following types of information available:

- Network configuration information (all models)
- ISDN configuration information (ISDN models)
- *Trunk configuration information* on page 23 (TDM models)
- Signaling server configuration information (ISUP models that have a Signaling Server)
- ISUP configuration information (basic ISUP audio and basic ISUP video models)

Network configuration information

The following information is required for all VX Integrated Media Platform models, except where indicated:

Required information	Value
Domain name for the VX Integrated Media Platform	
DNS server IP addresses	
VX Integrated Media Platform Ethernet 0 IP address, subnet mask, and default server	
VX Integrated Media Platform Ethernet 1 IP address, subnet mask, and default server	
(All models except IP-only audio) Media board 0 IP address, subnet mask, and default server	
(All models that have two media boards) Media board 1 IP address, subnet mask, and default server, if present.	
(All models that have three media boards) Media board 2 IP address, subnet mask, and default server, if present.	
IP address of the application server where the VoiceXML index is hosted	
Port of the HTTP server for the application server where the VoiceXML index is hosted	

ISDN configuration information

If you are using the ISDN audio or ISDN video model, gather the following ISDN configuration information:

Required information	Value
ISDN protocol variant	
ISDN equipment type	

Trunk configuration information

The following information is required for all VX Integrated Media Platform models, except IP-only audio and IP-only video:

Required information	Value
CRC	☐ On ☐ Off
Line impedance	☐ DSX1 ☐ G703_75_OHM ☐ G703_120_OHM
Frame type	☐ ESF ☐ D4 ☐ CEPT
Line coding	☐ AMI ☐ B8ZS (T1) ☐ HDB3 (E1) ☐ AMI_ZCS (T1) ☐ AMI_BELL (T1) ☐ AMI_DS (T1) ☐ AMI_GTE (T1)
Label or identification used to physically identify each E1 trunk on media board 0	Voice trunks (can also carry signaling): Trunk 1: _____ Trunk 2: _____ Trunk 3: _____ Trunk 4: _____ Trunk 5: _____ Trunk 6: _____ Trunk 7: _____ Trunk 8: _____ Signaling only trunks: Trunk 9: _____ Trunk 10: _____

Required information	Value
(For implementations with two media boards) Label or identification used to physically identify each E1 trunk on media board 1	Voice trunks (can also carry signaling): Trunk 1: _____ Trunk 2: _____ Trunk 3: _____ Trunk 4: _____ Trunk 5: _____ Trunk 6: _____ Trunk 7: _____ Trunk 8: _____ Signaling only trunks: Trunk 9: _____ Trunk 10: _____
(For implementations with three media boards) Label or identification used to physically identify each E1 trunk on media board 2	Voice trunks (can also carry signaling): Trunk 1: _____ Trunk 2: _____ Trunk 3: _____ Trunk 4: _____ Trunk 5: _____ Trunk 6: _____ Trunk 7: _____ Trunk 8: _____ Signaling only trunks: Trunk 9: _____ Trunk 10: _____

Signaling server configuration information

The following information is required for ISUP models that have a Signaling Server:

Required information	Value
Signaling server IP address	
Signaling server circuit start value	

Required information	Value
Signaling server variant (switch type)	☐ ANSI88 ☐ ANSI92 ☐ ANSI95 ☐ ETSIV2 ☐ ETSIV3 ☐ ITU97 ☐ ITUBLUE ☐ ITUWHITE ☐ JNTT ☐ Q767
Name of the signaling server associated with each trunk. Values for CG Board 1 and CG Board 2 are used for implementations with multiple CG boards.	CG Board 0: Trunk 1: _____ Trunk 2: _____ Trunk 3: _____ Trunk 4: _____ CG Board 1: Trunk 1: _____ Trunk 2: _____ Trunk 3: _____ Trunk 4: _____ CG Board 2: Trunk 1: _____ Trunk 2: _____ Trunk 3: _____ Trunk 4: _____

ISUP configuration information

The following types of information are required for basic ISUP audio and basic ISUP video models:

- ISUP and MTP base configuration information
- Destination point code information
- Trunk/circuit association information
- MTP link information

ISUP and MTP base configuration information

Required information	Value
Node point code	
ISUP variant (switch type) for the node point code	☐ ANSI88 ☐ ANSI92 ☐ ANSI95 ☐ ETSIV2 ☐ ETSIV3 ☐ ITU97 ☐ ITUBLUE ☐ ITUWHITE ☐ JNTT ☐ Q767
MTP link type for the node point code	☐ ANSI ☐ ITU ☐ JNTT
Subservice field for the node point code	☐ INTERNATIONAL ☐ SPARE ☐ NATIONAL ☐ RESERVED
Are MTP links embedded in voice trunks?	
Should CICs be incremented for TTS?	
Destination point codes (DPCs) for circuits on each T1/E1 trunk, and whether the point codes are adjacent point codes	
Trunk/circuit associations DPC and timeslot 1 CIC associated with each trunk	

Destination point code information

The following information is required for each destination point code:

Required information	Value
DPC	
Is the DPC an adjacent point code?	

Trunk/circuit association information

The following information is required for each trunk/circuit association:

Required information	Value
Media board number	
Trunk number	
DPC	
Timeslot 1 CIC	

MTP link information

The following information is required for each MTP link:

Required information	Value
CG board/trunk number for the link	
Is the link enabled?	
Destination point code	
Signaling link code (SLC)	
Timeslot to be reserved for signaling (usually 16 for E1 trunks).	

Logging into the platform for the first time

The information in this topic assumes you have installed and cabled the VX Integrated Media Platform, as described in *Installing the Dialogic® Vision™ Server™ TIGI2U*.

The VX Integrated Media Platform is shipped from the manufacturer with the following default IP network configuration for the first Ethernet interface (eth0):

- IP address: 192.168.0.1
- Subnet mask: 255.255.255.0
- Gateway: None
- Host name: VISION

You can configure the platform to use a static IP address (recommended) or DHCP.

Configuring the platform to use a static IP address

To configure the platform to use a static IP address, follow these steps:

Step	Action
1	Assign IP address 192.168.0.2 to the computer that will access the Vision™ Console.
2	Connect the Vision™ Console computer to eth0 on the VX Integrated Media Platform either directly using a crossover cable, or connect through a standalone Ethernet hub or switch.
3	Enter the following URL from an Internet Explorer or Mozilla browser on the Vision™ Console computer: http://192.168.0.1 You can also enter this URL from the VX Integrated Media Platform itself, after connecting a mouse, keyboard, and monitor to the platform hardware.
4	Log into the Vision™ Console using the following information: • User: admin • Password: vision The Overview page appears.
5	Click Host IP information in the Configuration menu. The Host IP information page appears.
6	Change the value of the IP address field in the ETH0 section to the desired IP address. Then set appropriate values for the Subnet mask and server fields.
7	In the Hostname section, configure the server's unique hostname, and then enter DNS server addresses in the DNS servers section, if required.
8	Double check the host IP information, and click Submit to apply the changes. The Web console attempts to reconnect after 15 seconds. If the server is on a different IP subnet than the Vision™ Console computer, the connection fails.
9	To validate that the gateway IP address is correct, change the IP address of the Vision™ Console machine to match the IP address subnet of the VX Integrated Media Platform. Then access the Vision™ Console by entering the IP address used in Step 6.

Obtaining an IP address through DHCP

If you need to use DHCP for the platform IP address, follow these steps:

Step	Action
1	Set up the DHCP server so the hostname and IP address assigned to the VX Integrated Media Platform is predetermined, for example, based on the platform's MAC address.
2	Assign IP address 192.168.0.2 to the computer that will access the Vision™ Console.
3	Connect the Vision™ Console computer to eth0 on the VX Integrated Media Platform either directly using a crossover cable, or through a standalone Ethernet hub.
4	Enter the following URL from an Internet Explorer or Mozilla browser on the Vision™ Console computer: http://192.168.0.1 You can also enter this URL from the VX Integrated Media Platform itself after connecting a mouse, keyboard, and monitor to the platform hardware.
5	Log into the VX Integrated Media Platform using the following information: • User: admin • Password: vision The Overview page appears.
6	Click Host IP information in the Configuration menu. The Host IP information page appears.
7	Select Enable DHCP in the DHCP setting field.
8	Click Submit .
9	Connect the VX Integrated Media Platform to the network.
10	Once the VX Integrated Media Platform has acquired its IP address and is reachable through a ping, access the Vision™ Console.
11	Click Services on the Operations menu.
12	Click Reboot , and wait for the system to restart before continuing.

Accessing the Vision Console

Use the Vision™ Console to configure and manage the VX Integrated Media Platform. To access the Vision™ Console, follow these steps:

Step	Action
1	Open one of the following local browsers. It is helpful to view the pages in full screen mode. • Mozilla 1.0 or later • Internet Explorer 6.0 or later
2	If this is the first time you are accessing the Vision™ Console, you must set up a host IP address for it as described in <i>Logging into the platform for the first time</i> on page 28. Once you have configured the host IP address for the Vision™ Console, enter the configured address.
3	Enter the following information: • User name (root, admin, or user) • Password (vision) The console displays the Overview page, which contains information about the servers, media boards, port rating, software version, installed patches, and licensing information for the VX Integrated Media Platform. This page also shows whether conferencing is enabled.

The user names for the Vision™ Console have the following rights:

User name	Associated rights
root	Super-user that can change the VX Integrated Media Platform model with assistance from Dialogic Services and Support. Note: Appropriate licenses are required to change the platform model.
admin	Administrator. This user can do everything an end user can do. In addition, this user can change all VX Integrated Media Platform configuration and provisioning settings, and can perform a port capacity upgrade. Note: Appropriate licenses are required to perform a port capacity upgrade.
user	End user. This user can monitor and add routes, and monitor and add CCXML applications. The options in the Configuration and Operations menus are read-only for this user.

Creating or revising a configuration

This topic describes how to create or revise a VX Integrated Media Platform configuration using the Vision™ Console. For information about configuration parameters, see the Vision™ Console parameters section.

To create a platform configuration using the Vision™ Console, follow these steps:

Step	Action
1	Access the Vision™ Console, and log in as admin. The Overview page appears with information about the version of the VX Integrated Media Platform.
2	Click Host IP information on the Configuration menu, and fill in the displayed field values. Click Submit .
3	Click Resources on the Configuration menu, and fill in the displayed field values. Click Submit .
4	Enter PSTN parameters, if you are configuring an ISDN audio, ISDN video, or ISUP video mode; or an ISUP audio model with signaling server integration. Click PSTN on the Configuration menu, and fill in the displayed field values. Click Submit .
5	Click Services on the Operations menu, and then click Restart all . Once the status of all services is STARTED, you can proceed.
6	Depending on the server model you are configuring, you might also need to manually specify additional configuration settings, as described in <i>Additional configuration tasks</i> on page 31.

To revise a server configuration using the Vision™ Console, follow these steps:

Step	Action
1	Access the Vision™ Console, and log in as admin: The Overview page appears with information about the version of the VX Integrated Media Platform.
2	Make the appropriate parameter changes.
3	If you changed any Configuration menu parameters, restart all services. Click Services on the Operations menu, and then click Restart all . Once the status of all services is STARTED, you can proceed.

Additional configuration tasks

The following table lists additional configuration tasks you may need to perform after using the Vision™ Console, depending on the configured environment:

Models	Configuration task	For more information, see...
All	Add additional settings for trunks in the <i>board-0.cfg</i> , <i>board-1.cfg</i> , or <i>board-2.cfg</i> files.	<i>Fine tuning the media board configuration</i> on page 123.
	Add additional settings for trunks and routes in the <i>telecom.conf</i> file. You can also specify telecom settings in the Options page on the Configuration menu.	<i>Fine tuning the telecom configuration</i> on page 127.
ISUP audio and ISUP	Add additional settings for SS7 signaling trunk in the <i>/opt/nmstx/etc/cx/txcfg1.txt</i> file.	

video models	Add TX IP information required for SIGTRAN in <i>/opt/nmstx/etc/cx/ipcfg1.txt</i> .	
	Add MTP3, SIGTRAN, and ISUP configuration information in <i>/opt/hs-data/raid/nms_hearsay/cfg/oam/ss7_config_default.xml</i> .	<i>Dialogic® Vision™ Signaling Server Administration Manual</i>

Backing up a configuration

To back up an existing VX Integrated Media Platform configuration, follow these steps:

Step	Action
1	Access the Vision™ Console, and log in as admin.
2	Click Import/Export on the Configuration menu. The Import/Export configuration page appears.
3	Click Download . The File Download window appears.
4	Click Save , locate the directory where you want to store the downloaded configuration, and enter the file name in the File name field. The configuration is downloaded to a .zip file.
5	Click Save . The system backs up the configuration and displays a message.
6	Click OK .

Restoring a configuration

To restore a VX Integrated Media Platform configuration, follow these steps:

Step	Action
1	Access the Vision™ Console, and log in as admin.
2	Click Import/Export in the Configuration menu. The Import/Export page appears.
3	Click Browse , and then locate the configuration you want to restore.
4	Click Apply . A confirmation message displays.
5	Click OK . The system restores the configuration.
6	Click Refresh to display the Overview page.
7	Click Services on the Operations menu, and then click Restart all to restart the VX Integrated Media Platform services.

Accessing the platform using a secure shell

The VX Integrated Media Platform is normally managed through the Vision™ Console. However, in some situations (for example, to take actions requested by Dialogic Services and Support), you may need to use a secure shell (ssh) to log into the platform.

To use ssh to log into the platform, follow these steps:

Step	Action
1	Use an ssh client to establish a secure shell connection. For example: <pre>ssh root@192.168.0.1</pre> Replace 192.168.0.1 with the IP address of your platform.
2	When prompted, enter the root password: vision. You now have access to a standard bash shell.

4

Vision™ Console parameters

Configuration menu parameters

The Configuration menu contains the following pages:

- Overview
- Host IP information
- Resource configuration
- SIP/RTP parameters
- Trunks
- PSTN
- Signaling Server
- Options
- Port upgrade
- Import/Export configuration

Overview

The Overview page displays the following information about the current VX Integrated Media Platform configuration:

- Model type and version
- Media board information
- Port rating
- Whether conferencing is enabled
- License information

Host IP information

The Host IP information page configures the IP network settings for the VX Integrated Media Platform. Access this page by clicking **Host IP information** on the Configuration menu.

The Host IP information page contains the following parameters:

Group	Parameter	Description
ETH0	Status	Indicates whether the first Ethernet interface on the VX Integrated Media Platform is activated when the platform starts up.
	DHCP settings	Indicates whether DHCP is enabled for the first Ethernet interface.
	IP address	IP address for the first Ethernet interface on the VX Integrated Media Platform, if DHCP is not enabled.
	Subnet mask	Subnet mask for the first Ethernet interface on the VX Integrated Media Platform, if DHCP is not enabled.
	Gateway	IP address of the router for the VX Integrated Media Platform network card, if DHCP is not enabled.
ETH1	Status	Indicates whether the second Ethernet interface on the VX Integrated Media Platform is activated when the server starts up.
	DHCP settings	Indicates whether DHCP is enabled for the second Ethernet interface.
	IP address	IP address for the second Ethernet interface on the VX Integrated Media Platform, if DHCP is not enabled.
	Subnet mask	Subnet mask for the second Ethernet interface on the VX Integrated Media Platform, if DHCP is not enabled.
	Gateway	IP address of the default gateway for the VX Integrated Media Platform network card.
Hostname	Hostname	Host name for the VX Integrated Media Platform.
DNS Servers	Server #1 - Server #3	IP address of the domain name servers for the VX Integrated Media Platform.

Resource configuration

The Resource configuration page determines whether conferencing and T.38 fax functionality is enabled. It also configures the size of codec and conferencing resource pools for the media boards in the VX Integrated Platform.

Access the Resource configuration page by clicking **Resources** on the Configuration menu. The Resource configuration page automatically opens in basic mode. By default, all codecs support RFC 2833 encoding, decoding, and DTMF detection.

Group	Associated implementations	Parameter	Description
Global resources	All	T.38	Indicates whether T.38 fax functionality is enabled.
	All	SIP info	Indicates whether the VX Integrated Media Platform can accept incoming SIP INFO messages with DTMF content.
	All	SIP to SIP server	Indicates whether the VX Integrated Media Platform can make SIP to SIP calls in passthrough mode; that is, with no audio transcoding.
Resource	All	G.711	Indicates whether G.711 mu-law or G.711 A-law is enabled.
	Implementations with a G.723 license from Dialogic	G.723	Indicates whether G.723 is enabled.
	All	G.726-32	Indicates whether G.726 is enabled.
	Implementations with a G.729 license from Dialogic	G.729	Indicates whether G.729 is enabled.
	Video models, or audio models with an AMR license from Dialogic	AMR	Indicates whether AMR is enabled.
	Video models	Mobile video	Indicates whether 3G-324M is enabled.
	Implementations with licensed conferencing ports	Conferencing	Indicates whether conferencing functionality is enabled.

Note: Do not use Advanced mode without contacting Dialogic Services and Support.

SIP/RTP parameters

The SIP/ RTP parameters page configures SIP-related settings. Access this page by clicking **SIP/RTP** on the Configuration menu.

Group	Associated implementations	Parameter	Description
Board #0	All	IP address	IP address of media board 0.
		Subnet mask	Subnet mask for media board 0.
		Gateway	IP address of the router for media board 0.
Board #1	Implementations where the VX Integrated Media Platform has two media boards	IP address	IP address of media board 1.
		Subnet mask	Subnet mask for media board 1.
		Gateway	IP address of the router for media board 1.
Board #2	Implementations where the VX Integrated Media Platform has three media boards	IP address	IP address of media board 2.
		Subnet mask	Subnet mask for media board 2.
		Gateway	IP address of the router for media board 2.
Audio codecs	All	Priority 1 - Priority 7	Order of the audio codecs presented in outbound SIP messages generated by the VX Integrated Media Platform. Priority 1 is the highest priority.
Video codecs	Video models	Priority 1 - Priority 4	Order of the video codecs presented in outbound SIP messages generated by the VX Integrated Media Platform. Priority 1 is the highest priority.

Trunks

The Trunks page configures trunk settings and circuit groups. Access this page by clicking **Trunks** on the Configuration menu.

Group	Associated Implementations	Parameter	Description
Trunk configuration	All	Frame type	Indicates frame type of T1 or E1.
		CRC signal checking	Indicates whether the media resource provides CRC signal checking.
		Line impedance	Type of cable connecting the media resource to the T1 or E1 network.
		Trunk framing	Framing format.
		Line code	Ones density maintenance method used on the trunk line to maintain a clear channel transmission.
PSTN routes	All	Route 1 - Route <i>n</i>	Defines the route circuits reserved for outgoing PSTN calls, and specifies how the circuits are selected. For a description of the valid values, see <i>Values for the PSTN routes group</i> on page 42. Click Add to reserve another route circuit for outgoing PSTN calls.
Board #0 trunks	All	Trunk 1 - Trunk <i>n</i>	Whether the specified trunk on Board 0 is enabled or disabled, and the PSTN route associated with the trunk.
Board #1 trunks	Implementations where the VX Integrated Media Platform has two media boards	Trunk 1 - Trunk <i>n</i>	Whether the specified trunk on Board 1 is enabled or disabled, and the PSTN route associated with the trunk.
Board #2 trunks	Implementations where the VX Integrated Media Platform has three media boards	Trunk 1 - Trunk <i>n</i>	Whether the specified trunk on Board 2 is enabled or disabled, and the PSTN route associated with the trunk.

Values for the PSTN routes group

Use a circuit code identifier (CCI) to identify a PSTN route. A CCI is an integer ranging from 1 to the highest circuit (CCI max). For example, if the routes contain four E1 trunks (each containing 30 circuits), circuits are numbered from 1 to 120. The value of CCI max is 120.

The following table describes the valid values for each route in the PSTN routes group:

Value	Description
FROM_TOP	(Default) Selects the first idle circuit in decreasing CCI order. This strategy always selects the highest available circuit. Example With this circuit selection strategy, a series of calls might be placed as follows: 1. A first call is placed on the last circuit, CCI max. 2. A second call is placed on circuit (CCI max – 1), because CCI max is busy processing the first call. 3. The first call terminates, so CCI max becomes idle. 4. A third call is placed on CCI max, because CCI max is now available. 5. A fourth call is placed on (CCI max – 2), because both CCI max and (CCI max – 1) are busy processing calls 3 and 2, respectively.
FROM_BOTTOM	Selects the first idle circuit in increasing CCI order. This strategy always selects the lowest available circuit. Example With this circuit selection strategy, a series of calls might be placed as follows: 1. A first call is placed on the first circuit, CCI 1. 2. A second call is placed on the second circuit, CCI 2, because the first circuit is busy processing the first call. 3. The first call terminates, so the first circuit becomes idle. 4. A third call is placed on CCI 1, because CCI 1 is now available. 5. A fourth call is placed on CCI 3, because CCI 1 and CCI 2 are busy processing calls 3 and 2, respectively.
DESCENDING	Selects a circuit by rotating circuits in decreasing CCI order, from the highest circuit (CCI max) down to the middle of the route $((CCI\ max / 2) + 1)$. If no circuit is idle on the second half of the route, a circuit on the first half of the route is selected. Example With this circuit selection strategy, a series of calls might be placed as follows: 1. A first call is placed on the last circuit, CCI max. 2. A second call is placed on (CCI max – 1). 3. The first call terminates, so CCI max becomes idle. 4. A third call is placed on (CCI max – 2). 5. For each subsequent call, the next lower circuit is selected up to the middle of the route. When the last circuit in the half route is reached $((CCI\ max / 2) + 1)$, the selection strategy rotates back to the last circuit CCI max, because that is the first available circuit in decreasing order of CCI.

Value	Description
ASCENDING	Selects a circuit by rotating circuits in increasing CCI order, from the first circuit (CCI 1) up to the middle of the route ($CCI\ max / 2$). If no circuit is idle on the first half of the route, a circuit on the second half is selected. Example With this circuit selection strategy, a series of calls might be placed as follows: 1. A first call is placed on the first circuit, CCI 1. 2. A second call is placed on the second circuit, CCI 2. 3. The first call terminates, so the first circuit becomes idle. 4. A third call is placed on CCI 3. 5. For each subsequent call, the next higher circuit is selected, up to the middle of the route. When the last circuit in the half route is reached ($CCI\ max / 2$), the selection strategy rotates back to CCI 1, because that is the first available circuit in increasing order of CCI.
TIMER	The selected circuit is the one on which the inactivity timer is the most important. At the beginning, all circuits have the same inactivity timer. The circuits are selected in decreasing CCI order, starting from CCI max down to 1. When all circuits have been used once, they are selected by the inactivity timer.

PSTN

The PSTN page contains additional configuration settings for the following VX Integrated Platform models:

- ISDN models
- ISUP models

Access this page by clicking **PSTN** on the Configuration menu.

Additional settings for ISDN models

The following settings apply to the ISDN audio and ISDN video models:

Group	Parameter	Description
ISDN	ISDN type	ISDN protocol variant.
	ISDN equipment	ISDN equipment type.

For more information, see *ISDN models* on page 12.

Additional settings for ISUP models

The following settings apply to the ISUP audio and ISUP video models:

Group	Parameter	Description
Signaling servers	ID	Signaling server ID.
	Redundant pair	If checked, the signaling server is part of a redundant pair.
	IP	IP address for the signaling server or the signaling server pair.
	Circuit start	Starting number of the circuit that the signaling server or signaling server pair can handle.
	Switch type	ISUP protocol variant for the signaling server or signaling server pair.
	Point code	Point code for the signaling server or signaling server pair. Specify the point code value as a decimal or hexadecimal number.
	Edit	Click Remove to remove the associated signaling server from the list.
Trunk - Signaling Server association	Trunk	T1 or E1 trunk handled by the previously defined signaling servers. The values in this field vary, depending on the number of boards and the trunk configuration.
	Signaling Server	Signaling server ID.

For more information, see *ISUP models* on page 14.

Signaling Server

The Signaling Server page contains additional configuration information for ISUP models that have a Signaling Server. The information on this page varies with the configuration, such as redundant pair configuration, and MTP or SIGTRAN transport protocol. Access this page by clicking **Signaling Server** on the Configuration menu.

Group	Associated implementations	Parameter	Description
Signaling Server redundant pair configuration	All ISUP	Redundant pair	If checked, the signaling server is part of a redundant pair.
		Redundant pair IP	IP address for the partner signaling server.
		Server Role	SS701 is the default primary server. SS702 is the default backup server.
TX Board Redundant IP information	Implementations with redundant servers	IP address	IP address of the TX board network interface used for redundancy.
		Subnet mask	Subnet mask of the TX board network interface used for redundancy.
		Mate IP address	IP address of the other TX board used in the redundancy pair.
ISUP & MTP base configuration	All ISUP	Node point code	Point code for the signaling server or signaling server pair. Specify the point code value as a decimal or hexadecimal number.
		Local routing context	Used when transport is set to SIGTRAN.
		SS7 subservice field	MTP 3 subservice.
ISUP base configuration	All ISUP	ISUP switch type	ISUP protocol variant for the Signaling Server or Signaling Server pair.
MTP base configuration	Implementations where transport is MTP	Transport	Indicates transport protocol: MTP or SIGTRAN.
		MTP link type	MTP 3 protocol variant.
		MTP links	Indicates how MTP links for ISUP arrive at the gateway: - When checked, the links are embedded in the trunks connected to the media boards. - When unchecked, the links are connected directly to the signaling board.
MTP base configuration	All ISUP	Transport	Indicates transport protocol: MTP or SIGTRAN.
		Network appearance code	Values are determined and configured by network operators on each side of an association.

Group	Associated implementations	Parameter	Description
		Service variant	Protocol variant of the M3UA service user.
		Source port	Listening STCP port.
TX Board SIGTRAN IP information	Implementations where transport is SIGTRAN	IP address	IP address of the TX board.
		Subnet mask	Subnet mask of the TX board.
		Gateway	IP address of the router for the TX board.
Peer signaling process	Implementations where transport is SIGTRAN	ID	ID of the peer signaling process.
		IP	IP address of the peer signaling process.
		Port	Remote SCTP port.
		PS type	Peer signaling type: IPSP or SGP.
		IPSP mode	Valid when PS type is IPSP. Indicates whether the IPSP mode is single-ended or double-ended.
		Dynamic routing key management	Indicates whether this peer signaling process can send and receive dynamic routing key management (DRKM) messages.
		Use network appearance	Determines whether the optional network appearance parameter is included when communicating with the remote peer.
		ASP	Indicates whether an ASP identifier is required in sent and/or received ASPUP and ASPUP ACK (ASP Up Acknowledgement) messages.
		Client	For PS type of IPSP, checked indicates that associations are automatically initiated from this PSP. For PS type of IPSP, unchecked means associations are not initiated from this PSP. The other side is expected to initiate any associations.
Destination point codes	All ISUP	Edit	Removes the peer signaling process.
		DPC	Destination point code for a circuit.
		Adjacent	When checked, the specified point code is an adjacent point code.
		Add new DPC	Enter the destination point code for another circuit, and click Add New DPC to add another destination point code to the list.
ISUP circuits	All ISUP	Index	Circuit group index.

Group	Associated implementations	Parameter	Description
definition		Start circuit	Starting number of the circuit group that the signaling server or signaling server pair can handle.
		Start CIC	Starting CIC of the circuit group that the signaling server or signaling server pair can handle.
		Number of circuits	Number of circuits in the circuit group.
		DPC	Destination point code associated with the circuit group.
		Edit	Removes the current circuit definition.
MTP links	Implementations where transport is MTP	Trunk	T1 or E1 trunk used for signaling. The values in this field vary, depending on the number of boards and the trunk configuration.
		Status	If checked, indicates that the MTP link is enabled.
		DPC	Destination point code associated with each route.
		SLC	Signaling link code.
		Signaling timeslot	Timeslot on the signaling trunks to be reserved for signaling. For E1 line types, the value is usually 16.
Remote MTP links	Implementations with redundant servers	Index	Index of the link definition.
		Port Number	Remote TX Board port number (corresponds to the MTP link index configured on the remote server).
		SLC	Signaling link code of the remote link.
		DPC	Destination point code of the remote link.
		Edit	Removes the current remote link definition.

For more information, see *ISUP models* on page 14.

Options

The Options page contains global and advanced settings. Access this page by clicking **Options** on the Configuration menu.

Group	Parameter	Description
Global	Billing	If checked, billing is enabled.
Advanced Telecom	Parameters in drop-down list	Select a parameter to be configured from the drop-down list and click Add . The parameter is displayed in the window above and its value can be edited. For more information on these parameters, see <i>Overview of fine tuning the configuration</i> on page 123.

Port upgrade

The Port upgrade page lets you view the port capacity of the VX Integrated Platform. Access this page by clicking **Capacity** on the Configuration menu.

The Port upgrade page contains the following parameters:

Parameter	Description
VoiceXML Interpreter	Port capacity of the VoiceXML Interpreter.
Call Server	Port capacity of the Call Server.

Import/Export configuration

The Import/Export configuration page lets you back up and restore a VX Integrated Platform configuration. Access this page by clicking **Import/Export** on the Configuration menu. For more information, see *Backing up a configuration* on page 33 and *Restoring a configuration* on page 34.

Operations menu parameters

The Operations menu contains the following pages:

- Services
- Maintenance

Services

Use the Services page to start or stop VX Integrated Media Platform services, quiesce the Call Server or VoiceXML Interpreter, and set up services to start up automatically when the VX Integrated Media Platform starts up. Access this page by clicking **Services** on the Operations menu.

The basic Services page contains the following parameters.

Field	Description
Status	Indicates the server status.
Managed services	Indicates the number of managed services.
Started services	Indicates the number of services that have been started.

The advanced Services page provides more detail on each service. This page contains the following parameters. Each row in the table represents one service.

Field	Description
Service name	Name of the service.
Status	Indicates status of the service, such as started, starting, stopped, and quiesced (quiesced is for Call Server and VoiceXML Interpreter only). When the VX Integrated Media Platform is starting, stopped, or quiesced, it cannot accept new calls.
Managed	Specifies how the service starts. When selected, the service is started automatically on start-up. If not selected, you must start the service manually.
Tasks	One of the following actions: • Start: Starts the service. • Stop: Stops the service. • Restart: Stops and then restarts the service. • Quiesce: Stops the Call Server or VoiceXML Interpreter from processing new calls. Quiesce does not affect calls that are currently being processed.

Maintenance

Use the Maintenance page to change the log levels of VX Integrated Media Platform logs, clear the application data cache, deploy licenses, and install patches. Access this page by clicking **Maintenance** on the Operations menu.

Group	Field	Description
Logging settings	Service name	Name of the service.

Group	Field	Description
	Log level	Log level of the associated service. Valid values are as follows, in order of decreasing severity and increasing verbosity: • FATAL • ERROR • WARNING • INFO1 • INFO2 • INFO3 • INFO4 • INFO5
		Click Cache to clear the application data cache for the associated service.
Signaling server role (ISUP models)	Current status	Indicates whether the Signaling Server being managed is the primary or backup server.
	Action	Click Switch to change the current server role.
N/A	Deploy license	For information about deploying licenses, contact Dialogic Services and Support.
N/A	Install patch	For information about installing patches, contact Dialogic Services and Support.

Provisioning menu parameters

The Provisioning menu contains the following pages:

- Call routing table
- CCXML application configuration
- VoiceXML application configuration

Call routing table

The Call Routing Table page defines routes for the VX Integrated Media Platform using the gateway routing table. This table is used when the Call Server is configured to work as a gateway. Access this page by clicking **Gateway routes** on the Provisioning menu. For more information, see the *Dialogic® Vision™ CX Video Gateway Administration Manual*.

CCXML application configuration

The CCXML application configuration page defines custom CCXML applications to the VX Integrated Media Platform. Access this page by clicking **CCXML applications** on the Provisioning menu. For more information see *Managing CCXML applications* on page 65.

VoiceXML application configuration

The VoiceXML application configuration page defines VoiceXML applications to the VX Integrated Media Platform. Access this page by clicking **VoiceXML applications** on the Provisioning menu. For more information, see *Managing VoiceXML applications* on page 61.

Monitoring menu parameters

The Monitoring menu contains the following pages:

- Trunks
- CCXML statistics
- Call Server status
- VoiceXML view
- VoiceXML status
- SSML Processor view
- SSML Processor status
- Signaling Monitor

Trunks

The Trunks page displays a trunk monitoring chart. Access this page by clicking **Trunks** on the Monitoring menu.

Field	Description
Circuits	Total number of configured circuits.
Circuit states	Information about circuit states. Valid values: • Idle: Circuit not used • Busy • Out of service • Blocked • Unknown
Trunk states	Information about trunk (line) states. Valid values: • In sync • Yellow alarm • Blue alarm • Red alarm
Signaling	(ISUP models) Number of circuits reserved for signaling (raw circuits). The Vision™ Console displays this information as you hover over an unknown circuit in the chart.
Blocked	Information about blocked circuits, if any. The Vision™ Console displays this information as you hover over a blocked circuit in the chart. Valid values: • local-maintenance: Circuit is locally blocked for maintenance. • remote-maintenance: Circuit is remotely blocked for maintenance. • local-hardware: Hardware is locally blocked. • remote-hardware: Hardware is remotely blocked.

CCXML statistics

The CCXML statistics page displays a bar chart that represents the following statistics:

- Number of active CCXML sessions, connections, and dialogs.
- Maximum number of CCXML sessions, connections, and dialogs.

Access this page by clicking **CCXML statistics** on the Monitoring menu.

Call Server status

The Call Server status page displays status information for the Call Server. Access this page by clicking **Call Server status** on the Monitoring menu.

Group	Description												
Server information	Contains the following fields: <table> <tr> <th>Field</th><th>Description</th></tr> <tr> <td>Status</td><td> Status of the Call Server: - Loading Telecom: server is starting. - Running: server started and is accepting incoming calls. - Stopped: server is quiesced and rejects incoming calls. </td></tr> <tr> <td>Active SIP calls</td><td>Number of active SIP calls.</td></tr> <tr> <td>Active PSTN calls</td><td>Number of active PSTN calls.</td></tr> <tr> <td>Total calls</td><td>Total number of calls since process startup.</td></tr> <tr> <td>Max concurrent calls</td><td>Maximum number of concurrent calls since process startup.</td></tr> </table>	Field	Description	Status	Status of the Call Server: - Loading Telecom: server is starting. - Running: server started and is accepting incoming calls. - Stopped: server is quiesced and rejects incoming calls.	Active SIP calls	Number of active SIP calls.	Active PSTN calls	Number of active PSTN calls.	Total calls	Total number of calls since process startup.	Max concurrent calls	Maximum number of concurrent calls since process startup.
Field	Description												
Status	Status of the Call Server: - Loading Telecom: server is starting. - Running: server started and is accepting incoming calls. - Stopped: server is quiesced and rejects incoming calls.												
Active SIP calls	Number of active SIP calls.												
Active PSTN calls	Number of active PSTN calls.												
Total calls	Total number of calls since process startup.												
Max concurrent calls	Maximum number of concurrent calls since process startup.												
System information	Contains the following fields: <table> <tr> <th>Field</th><th>Description</th></tr> <tr> <td>Version</td><td>Name and version of the Call Server, with the process name in brackets.</td></tr> <tr> <td>Startup time</td><td>Time the process started in coordinated universal time (UTC).</td></tr> <tr> <td>Uptime</td><td>Total time in days, hours, and minutes since process startup.</td></tr> <tr> <td>Channel</td><td>Configured number of channels accepting calls, available to place calls, or both.</td></tr> <tr> <td>Licenses</td><td>Available number of Call Server licenses.</td></tr> </table>	Field	Description	Version	Name and version of the Call Server, with the process name in brackets.	Startup time	Time the process started in coordinated universal time (UTC).	Uptime	Total time in days, hours, and minutes since process startup.	Channel	Configured number of channels accepting calls, available to place calls, or both.	Licenses	Available number of Call Server licenses.
Field	Description												
Version	Name and version of the Call Server, with the process name in brackets.												
Startup time	Time the process started in coordinated universal time (UTC).												
Uptime	Total time in days, hours, and minutes since process startup.												
Channel	Configured number of channels accepting calls, available to place calls, or both.												
Licenses	Available number of Call Server licenses.												
CCXML statistics	Displays the following statistics: - Number of active CCXML sessions, connections, and dialogs. - Maximum number of CCXML sessions, connections, and dialogs.												
Cache information	Displays the current and maximum memory usage and disk usage of the Call Server's internal caches. The server has the following types of caches: - CCXML, for CCXML scripts. - Script, for JavaScript files fetched from a CCXML script. When the limit is reached on a cache, older and less frequently used resources are deleted from the cache.												
Channel information	Displays the current execution (or health) status for each channel belonging to the server.												

VoiceXML Interpreter view

The VoiceXML Interpreter view page displays a real-time view of the VoiceXML Interpreter's current activity. Access this page by clicking **VoiceXML view** on the Monitoring menu.

Field	Description
[online] [offline]	Status of the VoiceXML Interpreter: • [online] if the channels are accepting new connections. • [offline] if the channels have been quiesced. For information on quiescing and restarting channels, see <i>Working with VX Integrated Media Platform services</i> on page 59.
calls in progress	Number of calls currently in progress.
calls total	Total number of calls since startup.
max concurrent calls	Maximum number of concurrent calls since startup.

The VoiceXML Interpreter view page also lists the channels as they are used, including the DNIS and ANI of each call.

VoiceXML Interpreter status

The VoiceXML Interpreter status page displays VoiceXML Interpreter status information. Access this page by clicking **VoiceXML status** on the Monitoring menu.

Group	Description										
Session information	Contains the following fields: <table> <tr> <th>Field</th><th>Description</th></tr> <tr> <td>Status</td><td> Status of the VoiceXML Interpreter: • [online] The channels are accepting new connections. • [offline] The channels are quiesced. </td></tr> <tr> <td>Calls in progress</td><td>Current number of calls.</td></tr> <tr> <td>Total calls</td><td>Total number of calls since startup.</td></tr> <tr> <td>Max concurrent calls</td><td>Maximum number of concurrent calls since process startup.</td></tr> </table>	Field	Description	Status	Status of the VoiceXML Interpreter: • [online] The channels are accepting new connections. • [offline] The channels are quiesced.	Calls in progress	Current number of calls.	Total calls	Total number of calls since startup.	Max concurrent calls	Maximum number of concurrent calls since process startup.
Field	Description										
Status	Status of the VoiceXML Interpreter: • [online] The channels are accepting new connections. • [offline] The channels are quiesced.										
Calls in progress	Current number of calls.										
Total calls	Total number of calls since startup.										
Max concurrent calls	Maximum number of concurrent calls since process startup.										

Group	Description																
System information	Contains the following fields: <table> <tr> <th>Field</th><th>Description</th></tr> <tr> <td>Version</td><td>Name and version of the VoiceXML Interpreter, with the process name in brackets.</td></tr> <tr> <td>Configuration file</td><td>Name of the configuration file for the current process.</td></tr> <tr> <td>MIO</td><td>Name and version of the current Media Interface Object implementation. The Media Interface Object is used by the Media Resource Function component of the VoiceXML subsystem.</td></tr> <tr> <td>Startup time</td><td>Time the process started in coordinated universal time (UTC).</td></tr> <tr> <td>Uptime</td><td>Total time in days, hours, and minutes since process startup.</td></tr> <tr> <td>Channels</td><td>Configured number of channels accepting calls, available to place calls, or both.</td></tr> <tr> <td>Licenses</td><td>Available number of licenses for the different call modes.</td></tr> </table>	Field	Description	Version	Name and version of the VoiceXML Interpreter, with the process name in brackets.	Configuration file	Name of the configuration file for the current process.	MIO	Name and version of the current Media Interface Object implementation. The Media Interface Object is used by the Media Resource Function component of the VoiceXML subsystem.	Startup time	Time the process started in coordinated universal time (UTC).	Uptime	Total time in days, hours, and minutes since process startup.	Channels	Configured number of channels accepting calls, available to place calls, or both.	Licenses	Available number of licenses for the different call modes.
Field	Description																
Version	Name and version of the VoiceXML Interpreter, with the process name in brackets.																
Configuration file	Name of the configuration file for the current process.																
MIO	Name and version of the current Media Interface Object implementation. The Media Interface Object is used by the Media Resource Function component of the VoiceXML subsystem.																
Startup time	Time the process started in coordinated universal time (UTC).																
Uptime	Total time in days, hours, and minutes since process startup.																
Channels	Configured number of channels accepting calls, available to place calls, or both.																
Licenses	Available number of licenses for the different call modes.																
Health information	Displays information about the health of the channels and identifies any channel that has generated an exception or is not responding. OK displays if all channels are operating normally.																
Cache information	Displays the current and maximum allowed sizes of the VoiceXML Interpreter's internal caches. The VoiceXML Interpreter has caches for: • VoiceXML documents • Grammar files • ECMA script files • XML documents When the limit is reached on a cache, older and less frequently used resources become eligible for deletion from the cache.																
Channel information	Displays the current execution (or health) status for each channel belonging to the VoiceXML Interpreter.																

SSML Processor view

The SSML Processor view page displays a real-time view of the SSML Processor's current activity. Access this page by clicking **SSML view** on the Monitoring menu.

Field	Description
[online] [offline]	Status of the SSML Processor: • [online] The channels are accepting new connections. • [offline] The channels are quiesced. For information about quiescing and restarting channels, see <i>Working with VX Integrated Media Platform services</i> on page 59.
current client connections	Number of calls currently in progress.
max concurrent client connections	Maximum number of concurrent calls since start-up.

The SSML Processor view page also displays:

- Information about each client as it connects and disconnects, including the client IP address.
- Text-to-speech (TTS) information including the vendor name, locale name, server address, server port number, and channel number.

SSML Processor status

This SSML Processor status page displays SSML Processor status information. Access this page by clicking **SSML status** on the Monitoring menu.

Group	Description														
Session information	Displays the following types of session information: <table> <tr> <th>Field</th><th>Description</th></tr> <tr> <td>Status</td><td> Status of the SSML Processor: • [online] The channels are accepting new connections. • [offline] The channels are quiesced. </td></tr> <tr> <td>Current client connections</td><td>Current number of connections.</td></tr> <tr> <td>Max concurrent client connections</td><td>Maximum number of concurrent client connections since process startup.</td></tr> </table>	Field	Description	Status	Status of the SSML Processor: • [online] The channels are accepting new connections. • [offline] The channels are quiesced.	Current client connections	Current number of connections.	Max concurrent client connections	Maximum number of concurrent client connections since process startup.						
Field	Description														
Status	Status of the SSML Processor: • [online] The channels are accepting new connections. • [offline] The channels are quiesced.														
Current client connections	Current number of connections.														
Max concurrent client connections	Maximum number of concurrent client connections since process startup.														
System information	Displays the following types of system information: <table> <tr> <th>Field</th><th>Description</th></tr> <tr> <td>Version</td><td>Name and version of the SSML Processor, with the process name given in brackets.</td></tr> <tr> <td>Configuration file</td><td>Name of the configuration file for the current process.</td></tr> <tr> <td>Audio driver</td><td>Name and version of the current audio driver (AD) implementation.</td></tr> <tr> <td>Startup time</td><td>Time, in coordinated universal time (UTC), at which the process was started.</td></tr> <tr> <td>Uptime</td><td>Total time in days, hours, and minutes since process startup.</td></tr> <tr> <td>Channels</td><td>Total number of SSML Processor channels.</td></tr> </table>	Field	Description	Version	Name and version of the SSML Processor, with the process name given in brackets.	Configuration file	Name of the configuration file for the current process.	Audio driver	Name and version of the current audio driver (AD) implementation.	Startup time	Time, in coordinated universal time (UTC), at which the process was started.	Uptime	Total time in days, hours, and minutes since process startup.	Channels	Total number of SSML Processor channels.
Field	Description														
Version	Name and version of the SSML Processor, with the process name given in brackets.														
Configuration file	Name of the configuration file for the current process.														
Audio driver	Name and version of the current audio driver (AD) implementation.														
Startup time	Time, in coordinated universal time (UTC), at which the process was started.														
Uptime	Total time in days, hours, and minutes since process startup.														
Channels	Total number of SSML Processor channels.														
Health information	Displays information about the health of the channels and identifies any channel that has generated an exception or is not responding. OK displays if all channels are operating normally.														
Cache information	Displays the current and maximum allowed sizes of the prompt and lexicon caches. The memory cache size and the disk cache size are displayed.														
Statistics information	Provides statistical information about TTS engine usage, giving the total usage, maximum concurrent usage, and number of available instances for each locale. An administrator uses this information to determine if the number of instances available for a particular locale is sufficient for the current load.														

Signaling Monitor

The Signaling Monitor page displays status information for a Signaling Server if one is used. Access this page by clicking **Signaling Monitor** on the Monitoring menu.

Group	Field	Description
Signaling server status	Server ID	ID of the signaling server.
	IP	IP address for the signaling server.
	Role	Indicates whether the server acts in a primary or secondary role.
	Location	Indicates whether the server is local or remote.
	Status	Status of the signaling server.
Signaling links status	Link index	MTP link identifier.
	Server ID	ID of the signaling server owning this link.
	Status	MTP link status.

5

Managing the VX Integrated Media Platform

Working with VX Integrated Media Platform services

Use the Vision™ Console to set up auto-start for VX Integrated Media Platform services and to manually start, stop, and restart these services. The Vision™ Console also lets you quiesce the Call Server service, VoiceXML Interpreter service, or both. This forces the service to stop processing new calls, but does not affect any calls currently being processed.

To work with VX Integrated Media Platform services:

Step	Action										
1	Access the Vision™ Console, as described in <i>Accessing the Vision Console</i> on page 30.										
2	Click Services on the Operations menu. The Services page appears.										
3	Click the Managed field to change the auto-start capability for the associated service.										
4	Click one of the following buttons in the Tasks column for the entity you want to manually start, stop, quiesce, or restart: <table><tr><th>Button</th><th>Description</th></tr><tr><td>Start</td><td>Starts the associated service. The service status changes to Starting while the service starts, and to Started after it is fully started.</td></tr><tr><td>Stop</td><td>Stops the associated service. The service status changes to Stopped when the service is stopped. If you click Stop for the Call Server or VoiceXML Interpreter service, the service stops without completing the in-process calls.</td></tr><tr><td>Quiesce</td><td>(Call Server and VoiceXML Interpreter services only) Forces the service to stop processing new calls, but does not affect any calls currently being processed. Use this functionality to interrupt service and carry out maintenance operations on an active server without disturbing active calls. Once maintenance operations are complete, you can stop, restart, or unquiesce the quiesced services. Unquiesce allows the server to accept new calls without restarting.</td></tr><tr><td>Restart</td><td>Stops and then restarts the associated service. If you restart the Call Server or VoiceXML Interpreter service, the service stops without completing the in-process calls. Click Restart All to stop and restart all server services.</td></tr></table>	Button	Description	Start	Starts the associated service. The service status changes to Starting while the service starts, and to Started after it is fully started.	Stop	Stops the associated service. The service status changes to Stopped when the service is stopped. If you click Stop for the Call Server or VoiceXML Interpreter service, the service stops without completing the in-process calls.	Quiesce	(Call Server and VoiceXML Interpreter services only) Forces the service to stop processing new calls, but does not affect any calls currently being processed. Use this functionality to interrupt service and carry out maintenance operations on an active server without disturbing active calls. Once maintenance operations are complete, you can stop, restart, or unquiesce the quiesced services. Unquiesce allows the server to accept new calls without restarting.	Restart	Stops and then restarts the associated service. If you restart the Call Server or VoiceXML Interpreter service, the service stops without completing the in-process calls. Click Restart All to stop and restart all server services.
Button	Description										
Start	Starts the associated service. The service status changes to Starting while the service starts, and to Started after it is fully started.										
Stop	Stops the associated service. The service status changes to Stopped when the service is stopped. If you click Stop for the Call Server or VoiceXML Interpreter service, the service stops without completing the in-process calls.										
Quiesce	(Call Server and VoiceXML Interpreter services only) Forces the service to stop processing new calls, but does not affect any calls currently being processed. Use this functionality to interrupt service and carry out maintenance operations on an active server without disturbing active calls. Once maintenance operations are complete, you can stop, restart, or unquiesce the quiesced services. Unquiesce allows the server to accept new calls without restarting.										
Restart	Stops and then restarts the associated service. If you restart the Call Server or VoiceXML Interpreter service, the service stops without completing the in-process calls. Click Restart All to stop and restart all server services.										

Once the platform is in production, you should stop components only when you need to troubleshoot a call issue. You should stop components in the following order:

1. Call Server
2. VoiceXML Interpreter
3. SSML Processor

Always quiesce the Call Server and VoiceXML Interpreter before you stop them, so that existing calls are not disrupted.

Viewing platform information

Use the Vision™ Console to view information about VX Integrated Media Platform components and interfaces. The following table describes the procedures for viewing information in the Vision™ Console:

Component	Type of information	Procedure
Trunks (for PSTN models)	Illustration of trunk and circuit states, which shows: • The total number of circuits. • Whether a circuit is idle, busy, out of service, or blocked. • Whether a trunk is in sync or in an alarm state. If a trunk is in an alarm state, the Trunks page shows the alarm.	Click Trunks on the Monitoring menu. For field descriptions, see <i>Trunks</i> on page 53.
CCXML statistics	Number of active CCXML sessions, connections, connections, conferences, and dialogs.	Click CCXML statistics on the Monitoring menu. For field descriptions, see <i>CCXML statistics</i> on page 53.
Call Server	Server information, system information, CCXML statistics, cache information, and channel information.	Click Call Server status on the Monitoring menu. For field descriptions, see <i>Call Server status</i> on page 54.
VoiceXML Interpreter	Current activity	Click VoiceXML view on the Monitoring menu. For field descriptions, see <i>VoiceXML Interpreter view</i> on page 55.
	Status	Click VoiceXML status on the Monitoring menu. For field descriptions, see <i>VoiceXML Interpreter status</i> on page 55.
SSML Processor	Current activity	Click SSML view on the Monitoring menu. For field descriptions, see <i>SSML Processor view</i> on page 56.
	Status	Click SSML status on the Monitoring menu. For field descriptions, see <i>SSML Processor status</i> on page 57.

Managing VoiceXML applications

Use the Vision™ Console to specify the following properties for individual VoiceXML applications:

- A number range that maps to the application. Incoming calls within the specified range are processed by the application.
- The initial URI to use for an incoming call, based on the specified number range.
- Whether logging is enabled, and the name of the application log file.
- Whether call transfers are allowed.
- The maximum call duration.

To specify properties for an application, first add the application to the Vision™ Console.

This topic describes how to:

- Add a VoiceXML application to the Vision Console
- Remove a VoiceXML application from the Vision Console
- Modify a VoiceXML application definition

It also describes the pattern matching syntax used in application definitions.

Adding a VoiceXML application to the Vision Console

To add a VoiceXML application to the Vision™ Console, follow these steps:

Step	Action														
1	Access the Vision™ Console, as described in <i>Accessing the Vision Console</i> on page 30.														
2	Click VoiceXML applications on the Provisioning menu. The VoiceXML application configuration page appears. To define a default application for all incoming calls, use the .% pattern. If no other matches are found for an incoming call, the initial URI associated with the .% pattern is used.														
3	In the Add new application box, enter the following information for the new application: <table border="1"> <thead> <tr> <th>Field</th><th>Description</th></tr> </thead> <tbody> <tr> <td>Number range</td><td> Number range that maps to the application. The number range can be a combination of digits and wildcards. If the dialed number of an incoming call matches the number ranges of multiple applications, the call is matched to the most specific number range. For information about the syntax used for specifying number ranges, see <i>Pattern matching syntax</i> on page 64. </td></tr> <tr> <td>Initial URI</td><td> Initial URI to use for a call, based on the number range of the dialed number (DNIS). If the VX Integrated Media Platform receives a call whose DNIS does not match the number range for any of the configured applications, the caller is redirected to the URI specified in the DefaultInitialURI setting of the VoiceXML Interpreter configuration file. For more information, see the <i>Dialogic® Vision™ VoiceXML Administration Manual</i>. </td></tr> <tr> <td>Logging enabled</td><td>If checked, logging is performed for the application.</td></tr> <tr> <td>Log subdirectory</td><td>Subdirectory to which application logs are written, if logging is enabled.</td></tr> <tr> <td>Transfer allowed</td><td>If checked, call transfers are allowed for the application.</td></tr> <tr> <td>Max call duration</td><td>Maximum call duration allowed for the application.</td></tr> </tbody> </table>	Field	Description	Number range	Number range that maps to the application. The number range can be a combination of digits and wildcards. If the dialed number of an incoming call matches the number ranges of multiple applications, the call is matched to the most specific number range. For information about the syntax used for specifying number ranges, see <i>Pattern matching syntax</i> on page 64.	Initial URI	Initial URI to use for a call, based on the number range of the dialed number (DNIS). If the VX Integrated Media Platform receives a call whose DNIS does not match the number range for any of the configured applications, the caller is redirected to the URI specified in the DefaultInitialURI setting of the VoiceXML Interpreter configuration file. For more information, see the <i>Dialogic® Vision™ VoiceXML Administration Manual</i>.	Logging enabled	If checked, logging is performed for the application.	Log subdirectory	Subdirectory to which application logs are written, if logging is enabled.	Transfer allowed	If checked, call transfers are allowed for the application.	Max call duration	Maximum call duration allowed for the application.
Field	Description														
Number range	Number range that maps to the application. The number range can be a combination of digits and wildcards. If the dialed number of an incoming call matches the number ranges of multiple applications, the call is matched to the most specific number range. For information about the syntax used for specifying number ranges, see <i>Pattern matching syntax</i> on page 64.														
Initial URI	Initial URI to use for a call, based on the number range of the dialed number (DNIS). If the VX Integrated Media Platform receives a call whose DNIS does not match the number range for any of the configured applications, the caller is redirected to the URI specified in the DefaultInitialURI setting of the VoiceXML Interpreter configuration file. For more information, see the <i>Dialogic® Vision™ VoiceXML Administration Manual</i>.														
Logging enabled	If checked, logging is performed for the application.														
Log subdirectory	Subdirectory to which application logs are written, if logging is enabled.														
Transfer allowed	If checked, call transfers are allowed for the application.														
Max call duration	Maximum call duration allowed for the application.														
4	Click Add. The Vision™ Console adds the new definition to the top of the definition list.														

The following example shows two defined applications. The first application matches incoming calls that have four-digit numbers. The second application matches all incoming calls that do not have four-digit numbers.

Applications:

Number range	Initial URI	Logging enabled	Log subdirectory	Transfer allowed	Max call duration (s)	Statistics	Edit
9999	http://10.10.41.1	☒		☐	600	View	Update Remove
.*	http://10.10.41.1	☒		☐	600	View	Update Remove

Add New Application:

Number range	Initial URI	Logging enabled	Log subdirectory	Transfer allowed	Max call duration (s)	
		☒		☐	600	Add

Removing a VoiceXML application from the Vision Console

To remove an application from the Vision™ Console:

Step	Action
1	Access the Vision™ Console, as described in <i>Accessing the Vision Console</i> on page 30.
2	Click VoiceXML applications on the Provisioning menu. The VoiceXML application configuration page appears.
3	Locate the row that contains the application you want to remove, and click Remove .

Modifying a VoiceXML application definition

To modify a VoiceXML application definition, follow these steps:

Step	Action
1	Access the Vision™ Console, as described in <i>Accessing the Vision Console</i> on page 30.
2	Click VoiceXML applications on the Provisioning menu. The VoiceXML application configuration page appears.
3	Add the application to the Vision™ Console.
4	Fill in appropriate information for the application. For a description of the entry fields, see <i>Adding a VoiceXML application to the Vision Console</i> on page 62.
5	Click Update to save the changes.

Pattern matching syntax

The following table describes the syntax used to specify patterns for matching the dialed number (DNIS) for inbound calls:

Symbol	Description
.	Wildcard that matches any single digit. For example, 123.... matches any dialed string beginning with 123, plus exactly four additional digits.
[]	Range of digits. A consecutive range is indicated with a hyphen (-); for example, [5-7]. A non-consecutive range is indicated with a comma (,); for example, [5,8]. Hyphens and commas can be used in combination; for example, [5-7,9]. Note: Only single-digit ranges are supported. For example, [98-102] is invalid.
()	A pattern; for example, 408(555). Used in conjunction with the symbol ?, %, or +.
?	Preceding digit occurred zero or one time.
%	Preceding digit occurred zero or more times. This is similar to the asterisk (*) used in regular expression.
+	Preceding digit occurred one or more times.

The following table provides examples of destination patterns and how they are interpreted:

Pattern	Translation
408555.+	408555, followed by one or more wildcard digits. Indicates the string must contain at least 7 digits starting with 408555.
408555.%	408555, followed by zero or more wildcard digits. Indicates the string must contain at least 408555.
408555+	40855, followed by 5 repeated one or more times.
408555%	40855, followed by 5 repeated zero or more times. Any explicitly matching digit before the % symbol is not stripped off.
408555?	40855, followed by 5 repeated zero or one time. Any explicitly matching digit before the ? symbol is not stripped off.
	40855, followed by 5, 6, or 7, plus any digit repeated one or more times.
40855[5-7].%	40855, followed by 5, 6, or 7, plus any digit repeated zero or more times.
40855[5-7]+1234	40855, followed by 5, 6, or 7 repeated one or more times, followed by 1234.
408(555)+1234	408, followed by 555, which may repeat one or more times, followed by 1234.

Managing CCXML applications

Use the Vision™ Console to specify the following properties for individual CCXML applications:

- A number range that maps to the CCXML application. Calls within the specified range are processed by the application.
- The initial URI to use for an incoming call, based on the specified number range.
- The dialog server that processes the initial URI.
- The outbound routes that the CCXML application uses for the PSTN calls that it creates.

The Vision™ Console adds this information to the CCXML application definition file (*ccxmlappcfg.xml*).

Note: To specify properties for an application, you must first add the application to the Vision™ Console.

This topic describes:

- Default CCXML application
- Adding a CCXML application definition to the Vision Console
- Removing a CCXML application definition from the Vision Console
- Modifying a CCXML application definition
- The pattern matching syntax used to match the dialed numbers of inbound calls and to define outbound routes

Default CCXML application

The VX Integrated Media Platform is shipped with a default CCXML application called *inbound.ccxml*, which is defined on the Vision™ Console as a custom application. This application matches all incoming calls and routes outbound calls to route 0. With route 0, the route is chosen automatically according to the associated incoming call. By default, all incoming calls are handled by dialog server 127.0.0.1:5070.

If you need additional routing functionality, you can do any of the following:

- Redefine *inbound.ccxml* to the Vision™ Console.
- Create one or more new CCXML applications, and define those applications as custom applications on the Vision™ Console.
- Access the gateway routing table functionality by creating one or more server applications. For information, see the CX Video Gateway documentation.

Adding a CCXML application definition to the Vision Console

To add a CCXML application to the Vision™ Console, follow these steps:

Step	Action
1	Access the Vision™ Console, as described in <i>Accessing the Vision Console</i> on page 30.

Step	Action										
2	Click CCXML applications on the Provisioning menu. The CCXML application configuration page appears. By default, there is one CCXML application called <i>inbound.ccxml</i> defined for the VX Integrated Media Platform. This application is defined as a custom application. Gateway applications are used for routing when the Call Server is used as a gateway. For more information, see the <i>Dialogic® Vision™ CX Video Gateway Administration Manual</i>.										
3	To add a new CCXML application to the Vision™ Console, click New in the Custom applications section, and enter the following information: <table border="1"> <thead> <tr> <th>Field</th><th>Description</th></tr> </thead> <tbody> <tr> <td>Number range</td><td> Number range associated with the CCXML application specified by the Initial URI value. The number range can be a combination of numbers, alphabetic characters, and wildcards. The VX Integrated Media Platform checks if the dialed number of an incoming call matches this. If it finds a match, the corresponding CCXML application is executed. If the dialed number matches the number range of multiple applications, the call is matched to the most specific number range. For more information, see <i>Pattern matching syntax</i> on page 64. </td></tr> <tr> <td>Initial URI</td><td>Initial URI to use for an incoming call, based on the number range of the dialed number.</td></tr> <tr> <td>Dialog servers</td><td>List of dialog servers for the application. Separate each dialog server with a comma.</td></tr> <tr> <td>Outbound routes</td><td> List of outbound routes for the calls that the CCXML application creates. Separate each route with a comma. PSTN routes For a PSTN route, the syntax is: <pre>route-Route_Number[Priority]</pre> where: • Route_Number identifies a PSTN route, as defined in the <i>telecom conf</i> file. • Priority specifies the priority level for load balancing over the telecom routes. If Route_Number is set to 0, the route is chosen automatically according to the associated incoming call. SIP routes For a SIP route, the syntax is: <pre>IP_Address:Port[Priority]</pre> where: • IP_Address:Port is the IP address and port of a SIP route. • Priority the priority level for load balancing over SIP routes. For both PSTN and SIP routes, Priority is optional, and its value is relative from 0 (default) to any required level. The highest priority is expressed by the lowest value (typically 0). Load balancing is performed between routes with the same priority and starts by routes defined with the highest priority. Note: If you have outbound PSTN routes defined in the CCXML application, the VX Integrated Media Platform processes those routes before it processes the routes defined here. </td></tr> </tbody> </table>	Field	Description	Number range	Number range associated with the CCXML application specified by the Initial URI value. The number range can be a combination of numbers, alphabetic characters, and wildcards. The VX Integrated Media Platform checks if the dialed number of an incoming call matches this. If it finds a match, the corresponding CCXML application is executed. If the dialed number matches the number range of multiple applications, the call is matched to the most specific number range. For more information, see <i>Pattern matching syntax</i> on page 64.	Initial URI	Initial URI to use for an incoming call, based on the number range of the dialed number.	Dialog servers	List of dialog servers for the application. Separate each dialog server with a comma.	Outbound routes	List of outbound routes for the calls that the CCXML application creates. Separate each route with a comma. PSTN routes For a PSTN route, the syntax is: <pre>route-Route_Number[Priority]</pre> where: • Route_Number identifies a PSTN route, as defined in the <i>telecom conf</i> file. • Priority specifies the priority level for load balancing over the telecom routes. If Route_Number is set to 0, the route is chosen automatically according to the associated incoming call. SIP routes For a SIP route, the syntax is: <pre>IP_Address:Port[Priority]</pre> where: • IP_Address:Port is the IP address and port of a SIP route. • Priority the priority level for load balancing over SIP routes. For both PSTN and SIP routes, Priority is optional, and its value is relative from 0 (default) to any required level. The highest priority is expressed by the lowest value (typically 0). Load balancing is performed between routes with the same priority and starts by routes defined with the highest priority. Note: If you have outbound PSTN routes defined in the CCXML application, the VX Integrated Media Platform processes those routes before it processes the routes defined here.
Field	Description										
Number range	Number range associated with the CCXML application specified by the Initial URI value. The number range can be a combination of numbers, alphabetic characters, and wildcards. The VX Integrated Media Platform checks if the dialed number of an incoming call matches this. If it finds a match, the corresponding CCXML application is executed. If the dialed number matches the number range of multiple applications, the call is matched to the most specific number range. For more information, see <i>Pattern matching syntax</i> on page 64.										
Initial URI	Initial URI to use for an incoming call, based on the number range of the dialed number.										
Dialog servers	List of dialog servers for the application. Separate each dialog server with a comma.										
Outbound routes	List of outbound routes for the calls that the CCXML application creates. Separate each route with a comma. PSTN routes For a PSTN route, the syntax is: <pre>route-Route_Number[Priority]</pre> where: • Route_Number identifies a PSTN route, as defined in the <i>telecom conf</i> file. • Priority specifies the priority level for load balancing over the telecom routes. If Route_Number is set to 0, the route is chosen automatically according to the associated incoming call. SIP routes For a SIP route, the syntax is: <pre>IP_Address:Port[Priority]</pre> where: • IP_Address:Port is the IP address and port of a SIP route. • Priority the priority level for load balancing over SIP routes. For both PSTN and SIP routes, Priority is optional, and its value is relative from 0 (default) to any required level. The highest priority is expressed by the lowest value (typically 0). Load balancing is performed between routes with the same priority and starts by routes defined with the highest priority. Note: If you have outbound PSTN routes defined in the CCXML application, the VX Integrated Media Platform processes those routes before it processes the routes defined here.										

Step	Action
4	Click Apply . The Vision™ Console adds the new definition to the top of the definition list in the Custom applications section.

Removing a CCXML application definition from the Vision Console

To remove an application from the Vision™ Console, follow these steps:

Step	Action
1	Access the Vision™ Console, as described in <i>Accessing the Vision Console</i> on page 30.
2	Click CCXML applications in the Provisioning menu. The CCXML application configuration page appears.
3	Locate the row that contains the application you want to remove, and click Remove . A confirmation message appears.
4	Click OK to remove the application.

Modifying a CCXML application definition

To modify a CCXML application definition, follow these steps:

Step	Action
1	Access the Vision™ Console, as described in <i>Accessing the Vision Console</i> on page 30.
2	Click CCXML applications on the Provisioning menu. The CCXML application configuration page appears.
3	Click Edit . The Edit CCXML application page appears.
4	Modify the application as appropriate. For a description of the entry fields, see <i>Adding a CCXML application definition to the Vision Console</i> on page 65.
5	Click Apply to save the changes.

CCXML application definition pattern matching syntax

The following table describes the syntax used to specify patterns for matching the dialed number (DNIS) for inbound calls. This is based on standard regular expression syntax.

Symbol	Description
.	Wildcard that matches any single digit or character. For example, 123.... matches any dialed string beginning with 123, plus exactly four additional digits.
[]	Range of digits. A consecutive range is indicated with a hyphen (-); for example, [5-7]. A non-consecutive range is indicated with a comma (,); for example, [5,8]. Hyphens and commas can be used in combination; for example, [5-7,9]. Note: Only single-digit ranges are supported. For example, [98-102] is invalid.
()	A pattern; for example, 408(555). Used in conjunction with the symbol ?, *, or +.
?	Preceding digit occurred zero or one time.
* or %	Preceding digit occurred zero or more times.
+	Preceding digit occurred one or more times.

The following table provides examples of destination patterns and how they are interpreted:

Pattern	Translation
408555.+	408555, followed by one or more wildcard digits. Indicates the string must contain at least 7 digits starting with 408555.
408555.*	408555, followed by zero or more wildcard digits. Indicates the string must contain at least 408555.
408555+	40855, followed by 5 repeated one or more times.
408555*	40855, followed by 5 repeated zero or more times. Any explicitly matching digit before the * (asterisk) is not stripped off.
408555?	40855, followed by 5 repeated zero or one time. Any explicitly matching digit before the ? (question mark) is not stripped off.
40855[5-7].+	40855, followed by 5, 6, or 7, plus any digit repeated one or more times.
40855[5-7].*	40855, followed by 5, 6, or 7, plus any digit repeated zero or more times.
40855[5-7]+1234	40855, followed by 5, 6, or 7 repeated one or more times, followed by 1234.
408(555)+1234	408, followed by 555, which may repeat one or more times, followed by 1234.

6

Developing VoiceXML applications

Overview of developing VoiceXML applications

Use VoiceXML 2.1 to develop speech and video applications that interact with the VX Integrated Media Platform.

This section describes how to use VoiceXML to:

- Manage media
- Develop an audio application
- Develop a video application
- Use the record utterance functionality

Managing media

VoiceXML applications have file:// access to local files and to files that reside on NFS shares. They also have HTTP:// access to remote media files on web servers.

The HTTP:// access works as follows:

- All media files are fetched from a web server over HTTP or HTTPS.
- The VX Integrated Media Platform reads the MIME type of the media file as presented by the web server. If the MIME type is not available, the platform uses the filename extension to determine the media type of the file.
- The VX Integrated Media Platform implements and follows all caching parameters as presented in the HTTP headers for the media files.
- All announcement media files are available for playing as soon as they are installed in the correct location on the application server.
- Media files with a streamable format begin streaming to the bearer connection when the HTTP transfer starts. The media server does not wait for the entire media file to transfer.

Developing an audio application

With all models of the VX Integrated Media Platform, the application uses standard VoiceXML 2.1 elements to develop audio applications.

For information about standard VoiceXML elements, including their descriptions and syntax, see the *Dialogic® Vision™ VoiceXML Reference Manual*.

Developing a video application

The ISDN video, ISUP video, and IP-only video models of the VX Integrated Media Platform use VoiceXML extensions in the video application to play, record, and post video messages.

Playing a video message

Use the VoiceXML <audio> element to play a video message in a 3GP file.

```
<prompt>
  Message 1, received yesterday at 10:45 pm.
  <audio src="http://192.168.1.1/msg01.3gp"/>
  End of message.
</prompt>
```

Note: When a video clip finishes playing, the last frame remains on the screen.

Because video clips play synchronously within other audio markup, the mixing of audio and video clips is seamless. Video clips may or may not include audio.

For more information about the VoiceXML <audio> element, see the *Dialogic® Vision™ VoiceXML Reference Manual*.

The VX Integrated Media Platform supports multiple video codecs (H.263, MPEG-4, and H.264). Since video transcoding is not provided, the VoiceXML application must select a 3GP file that has the appropriate codec. Use the proprietary variable `session.connection.videostreamtype` in the file name or as part of the file path to select the file. For example:

```
<prompt>
  Message 1, received yesterday at 10:45 pm.
  Using session.connection.videostreamtype as part of the file name
  <audio expr="'http://192.168.1.1/msg01-'+session.connection.videostreamtype+'.3gp'"/>
  End of message.
</prompt>
or
<prompt>
  Message 1, received yesterday at 10:45 pm.
  Using session.connection.videostreamtype as part of the path
  <audio expr="'http://192.168.1.1/'+session.connection.videostreamtype+'/msg01.3gp'"/>
  End of message.
</prompt>
```

Recording and posting a video message

Use the VoiceXML <record> element to record a video clip. For example:

```
<record name="videomsg" beep="true" type="video/3gpp">
  <prompt timeout="5s">
    Record a video message after the beep.
  </prompt>
</record>
```

Then use the <submit> element as a MIME body in an HTTP POST request to send the message to a web server. For example:

```

<field name="confirm">
<grammar type="application/srgs+xml" src="/grammars/boolean.grxml"/>
<prompt>
  Here is your recorded video message. <audio expr="videomsg"/>
</prompt>
<prompt>
  To keep it, say yes. To discard it, say no.
</prompt>
<filled>
  <if cond="confirm">
    <submit next="save_message.pl" enctype="multipart/form-data" method="post"
    namelist="videomsg"/>
  </if>
  <clear/>
</filled>
</field>

```

Note: If the application allows multiple codecs, then every prompt must be encoded for each supported codec and saved to a separate 3GP file. For example, suppose the application supports H.263 and H.264, and it has one hello prompt. The application needs to create two 3GP files for the hello prompt: one for H.263 and one for H.264.

MPEG-4 limitation for 3G-324M handsets

When building video VoiceXML applications intended to be used by 3G-324M handsets, MPEG-4 has a known limitation. The MPEG-4 decoder configuration information (DCI) is negotiated as part of the initial 3G-324M call setup, and can only be changed by closing and re-negotiating a new video channel on the 3GP leg of the call. This procedure is not widely supported by 3G handsets.

Because of this limitation, you should consider the following options:

Action	Options
Playing a video clip	Encode all MPEG-4 3GP files using identical parameters. Use the associated DCI to configure the mpeg4DecoderInfo parameter for the Call Server. For more information, see the <i>Dialogic® Vision™ Call Server Administration Manual</i>.
Recording and posting a video message	Do not use MPEG-4 for these actions, because the DCI used on the handset may be incompatible with that used by the VX Integrated Media Platform to play back the file to the user.

Using the record utterance functionality

To use the record utterance functionality in a VoiceXML application, follow these steps:

Step	Action
1	Set the "recordutterance" property in the VoiceXML script where the simultaneous recording and recognition will occur. For example: <pre>: <property name="recordutterance" value="true"/> : <form id="myForm"> <field name="myField"> : </field> </form> :</pre>
2	After recognition, you can reference the recognized utterance by using the shadow variable "lastResult\$.recording". For example, the following sample script plays back the recognized utterance and submits it to an external server for storage: <pre>: <filled> <prompt>You just said <value expr="lastResult\$.recording"></prompt> <submit next="http://myserver.mydomain.com/persist_utterance.jsp" method="post" namelist="lastResult\$.recording" enctype="multipart/form-data"/> </filled> :</pre>

7

VX Integrated Media Platform logging

Overview of VX Integrated Media Platform logging

The VX Integrated Media Platform logs error codes and alarms and can report statistics for incoming and outgoing calls. The following VX Integrated Media Platform components generate log files:

- Call Server
- VoiceXML Interpreter
- SSML Processor
- VoiceXML application
- SSML application

This section also discusses *logging and performance* on page 89.

Call Server system logs

The Call Server creates system logs that record information about the active Call Server process. The name format for the log is:

```
callserver_creationdate_[index].log
```

where:

- **creationdate** is the date the log file was created, formatted as local server time by default.
- **index** is an integer specifying the current incremented Call Server system log file. This value is reset daily and incremented when the configured maximum system log file size is reached or when the Call Server is restarted. Because the index value is a timestamp, indexes are ordered, but not necessarily consecutive.

For example, the following Call Server system log file was generated by the callserver process on 2008-09-15. It has an index value of 4:

```
callserver_2008_09-15_004.log
```

This topic describes the following information about Call Server system logs:

- Logging levels
- Logging defaults
- Changing the logging level
- Changing other logging defaults
- Log file format

Logging levels

The following table describes the logging levels for Call Server system logs in decreasing severity and increasing verbosity order:

Logging level	Description
FATAL	Logs only critical errors.
ERROR	Logs all errors.
WARNING	Logs all errors and warnings.
INFO1	Logs information useful for first-level debugging and also logs some normal events.
INFO2	Logs information useful for second-level debugging.
INFO3	Because this level generates abundant log information, Dialogic recommends that you set this level only at the request of Dialogic Services and Support.
INFO4	The most verbose option, which logs low-level detailed information.
INFO5	Because this level generates abundant log information, Dialogic recommends that you set this level only at the request of Dialogic Services and Support.

Logging defaults

The default values for Call Server system logs are:

- The logging level is set to 1, which displays events encountered during typical Call Server operations.

- The log file is stored in the *vx/callserver/logs* directory.
- The maximum number of log files in the log directory is 50. If the log directory contains 50 log files, then the oldest log file is deleted when the 51st log file is added.
- The maximum size of a log file is 10 MB. If a log file reaches that size, a new log file is created.
- The date/time format for the log creation date and log entry date is in local server time.
- The index for the Call Server system log file is reset each day. The index is incremented when a new log file is started because the maximum system log file size was reached or the 51st log file was added to the log directory.

Changing the logging level

To change the Call Server system logging level, follow these steps:

Step	Action
1	Access the Vision™ Console, as described in <i>Accessing the Vision Console</i> on page 30.
2	Click Maintenance on the Operations menu. The Maintenance page appears.
3	Select the desired log level in the Log level field associated with the Call Server service, and click Submit .

Changing other logging defaults

The following table describes how to change the other logging defaults for the Call Server system log:

To change the...	Modify the...
Log file location	LogDir setting in the <i>callserver.conf</i> file.
Maximum number of log files in the specified directory	SystemLogFileMaxNum setting in the <i>callserver.conf</i> file.
Maximum size of the log file	SystemLogFileMaxSize setting in the <i>callserver.conf</i> file.
Date/time format of the log file	SystemLogTime setting in the <i>callserver.conf</i> file.

For more information, see the *Dialogic® Vision™ Call Server Administration Manual*.

Log file format

The format of each Call Server system log message is:

```
time-stamp [severity] [origin:code] [UID:threadID] (alarm) [message]
```

The following table describes the Call Server system log fields:

Field	Description
severity	Level of the log message. Valid values (in order of decreasing severity and increasing verbosity): • FATAL (a severe malfunction from which the server processor cannot recover) • ERROR • WARNING • INFO1 • INFO2 • INFO3 • INFO4 • INFO5 (highest/most verbose level of detail)
origin	Call Server component to which the log message refers.
code	Trace identifier of the message in the Call Server component to which the log message refers.
UID	Reserved for future use. The UID is represented by a hyphen (-) in the log file.
threadID	Identifier associated with the thread that generated the message. Use this field to track the progress of a single session or call when several requests are being processed simultaneously.
alarm	Optional field that is included only when the log message refers to an alarm notification. In this situation, the field describes the general category of the alarm. Alarm categories include: • Started • Quiesced • Shutdown • LicenseCheck • ConfigurationError • InitializationError • SoftwareException • InternalError • ResourceLimitation • CommunicationFailure • ProcessingFailure • InvalidArgument • UnexpectedEvent • NotificationDiscarded • Watchdog • Timeout
message	Text description of the logged occurrence.

VoiceXML Interpreter system logs

The VoiceXML Interpreter creates system logs that contains information about the active VoiceXML Interpreter process. The name format for the log is:

```
vxmliinterpreter_creationdate_[index].log
```

where:

- **creationdate** is the date the log file was created, formatted as local server time by default.
- **index** is an integer specifying the current incremented VoiceXML Interpreter system log file. This value is reset daily and incremented when the configured maximum system log file size is reached or when the VoiceXML Interpreter is restarted. Because the index value is a timestamp, indexes are ordered, but not necessarily consecutive.

For example, the following VoiceXML Interpreter system log file was generated by the vxmliinterpreter process on 2008-08-28. It has an index value of 5:

```
vxmliinterpreter_2008_08-28_005.log
```

This topic describes the following information about VoiceXML Interpreter system logs:

- Logging levels
- Logging defaults
- Changing the logging level
- Changing other logging defaults
- Log file format

Logging levels

The following table describes the logging levels for VoiceXML Interpreter system logs in decreasing severity and increasing verbosity order:

Logging level	Description
FATAL	Logs only critical errors.
ERROR	Logs all errors.
WARNING	Logs all errors and warnings.
INFO1	Logs information useful for first-level debugging and also logs some normal events.
INFO2	Logs information useful for second-level debugging.
INFO3	Because this level generates abundant log information, you should set this level only at the request of Dialogic Services and Support.
INFO4	The most verbose option, which logs low-level detailed information.
INFO5	Because this level generates abundant log information, you should set this level only at the request of Dialogic Services and Support.

Logging defaults

The default values for VoiceXML Interpreter system logs are:

- The logging level is set to INFO1. This is sufficient for routine logging on a production machine.
- The system log file is stored in the `vx/vxmlinterpreter/logs` directory.
- The maximum number of log files in the log directory is 50. If the log directory contains 50 log files, then the oldest log file is deleted when the 51st log file is added.
- The maximum size of a log file is 10 MB. When a log file reaches this size, a new log file is added.
- The date/time format for the log creation date and log entry date is in local server time.
- The index for the VoiceXML Interpreter system log file is reset each day. The index is incremented when a new log file is started because the maximum system log file size was reached or the 51st log file was added to the log directory.

For more information, see the *Dialogic® Vision™ VoiceXML Administration Manual*.

Changing the logging level

To change the VoiceXML Interpreter system logging level, follow these steps:

Step	Action
1	Access the Vision™ Console, as described in <i>Accessing the Vision Console</i> on page 30.
2	Click Maintenance on the Operations menu. The Maintenance page appears.
3	Select the desired log level in the Log level field associated with the VoiceXML Interpreter service, and click Submit .

Changing other logging defaults

The following table describes how to change the other logging defaults for the VoiceXML Interpreter system log:

To change the...	Modify the...
Log file location	LogDir setting in the <code>vxmlinterpreter.conf</code> file.
Maximum number of log files in the specified directory	SystemLogFileMaxNum setting in the <code>vxmlinterpreter.conf</code> file.
Maximum size of the log file	SystemLogFileMaxSize setting in the <code>vxmlinterpreter.conf</code> file.
Date/time format of the log file	SystemLogTime setting in the <code>vxmlinterpreter.conf</code> file.

For more information, see the *Dialogic® Vision™ VoiceXML Administration Manual*.

Log file format

The format of each VoiceXML Interpreter system log message is:

```
[timestamp] [severity] [category] [channel ID:thread ID] message
```

The following table describes the fields in a VoiceXML Interpreter system log message:

Field	Description																				
timestamp	Date and time of the log message, specified in local server time by default. Use the SystemLogTime setting in the <i>vxmlinterpreter.conf</i> file to change the time format to Greenwich Mean Time (GMT). This also changes the time format in the log file name. For information, see the <i>Dialogic® Vision™ VoiceXML Administration Manual</i> .																				
severity	Level of the log message. Valid values (in order of decreasing severity and increasing verbosity): - FATAL (a severe malfunction from which the VoiceXML Interpreter process cannot recover) - ERROR - WARNING - INFO1 (default) - INFO2 - INFO3 - INFO4 - INFO5 (highest/most verbose level of detail)																				
category	General category to which the message refers: <table border="1"> <thead> <tr> <th>Label</th><th>Description</th></tr> </thead> <tbody> <tr> <td>Xml Parser</td><td>XML parser-related message.</td></tr> <tr> <td>VoiceXmlInterpreter</td><td>VoiceXML Interpreter flow/form interpretation algorithm related message.</td></tr> <tr> <td>I/O</td><td>Recognition/speech synthesis-related message.</td></tr> <tr> <td>Debug</td><td>Debug message.</td></tr> <tr> <td>Telephony</td><td>Telephony-related message.</td></tr> <tr> <td>Database</td><td>Database-related message.</td></tr> <tr> <td>JS</td><td>ECMA/JavaScript-related message.</td></tr> <tr> <td>Billing</td><td>Billing message.</td></tr> <tr> <td>Grammar</td><td>Grammar-related message.</td></tr> </tbody> </table>	Label	Description	Xml Parser	XML parser-related message.	VoiceXmlInterpreter	VoiceXML Interpreter flow/form interpretation algorithm related message.	I/O	Recognition/speech synthesis-related message.	Debug	Debug message.	Telephony	Telephony-related message.	Database	Database-related message.	JS	ECMA/JavaScript-related message.	Billing	Billing message.	Grammar	Grammar-related message.
Label	Description																				
Xml Parser	XML parser-related message.																				
VoiceXmlInterpreter	VoiceXML Interpreter flow/form interpretation algorithm related message.																				
I/O	Recognition/speech synthesis-related message.																				
Debug	Debug message.																				
Telephony	Telephony-related message.																				
Database	Database-related message.																				
JS	ECMA/JavaScript-related message.																				
Billing	Billing message.																				
Grammar	Grammar-related message.																				
channel ID	ID of the VoiceXML Interpreter telephony channel that generated the message. Otherwise, the channel ID is replaced with a dash, such as for process-wide log messages.																				
thread ID	ID of the thread that generated the message. This can be used to track the progress of a single session in the log file when there are several requests being processed simultaneously.																				
message	Log message.																				

SSML Processor system logs

The SSML Processor creates system logs that contain information about the active SSML Processor process. The name format for the system log is:

```
ssmlprocessor_creationdate_[index].log
```

where:

- **creationdate** is the date the log file was created, formatted as local server time by default.
- **index** is an integer specifying the current incremented SSML Processor system log file. This value is reset daily and incremented when the configured maximum system log file size is reached or when the VoiceXML Interpreter is restarted. Because the index value is a timestamp, indexes are ordered, but not necessarily consecutive.

For example, the following SSML Processor system log file was generated by the vxmlinterpreter process on 2008-05-23. It has an index value of 5:

```
ssmlprocessor_2008_05-23_005.log
```

This topic describes the following information about SSML Processor system logs:

- Logging levels
- Logging defaults
- Changing the logging level
- Changing other logging defaults
- Log file format

Logging levels

The following table describes the logging levels for SSML Processor system logs in decreasing severity and increasing verbosity order:

Logging level	Description
FATAL	Logs only critical errors.
ERROR	Logs all errors.
WARNING	Logs all errors and warnings.
INFO1	Logs information useful for first-level debugging and also logs some normal events.
INFO2	Logs information useful for second-level debugging.
INFO3	Note: Because this level generates abundant log information, you should set this level only at the request of Dialogic Services and Support.
INFO4	The most verbose option, which logs low-level detailed information.
INFO5	Note: Because this level generates abundant log information, you should set this level only at the request of Dialogic Services and Support.

Logging defaults

The default values for SSML Processor system logs are:

- The logging level is set to INFO1. This is sufficient for routine logging on a production machine.

- The system log file is stored in the `vx/ssmlprocessor/logs` directory.
- The maximum number of log files in the log directory is 50. If the log directory contains 50 log files, then the oldest log file is deleted when the 51st log file is added.
- The maximum size of a log file is 10 MB. When a log file reaches this size, a new log file is added.
- The date/time format for the log creation date and log entry date is in local server time.
- The index for the SSML Processor system log file is reset each day. The index is incremented when a new log file is started because the maximum system log file size was reached or the 51st log file was added to the log directory.

For more information, see the *Dialogic® Vision™ VoiceXML Administration Manual*.

Changing the logging level

To change the SSML Processor system logging level, follow these steps:

Step	Action
1	Access the Vision™ Console, as described in <i>Accessing the Vision Console</i> on page 30.
2	Click Maintenance on the Operations menu. The Maintenance page appears.
3	Select the desired log level in the Log level field associated with the SSML Processor service, and click Submit .

Changing other logging defaults

The following table describes how to change the other logging defaults for the SSML Processor system log:

To change the...	Modify the...
Log file location	LogDir setting in the <code>ssmlprocessor.conf</code> file.
Maximum number of log files in the specified directory	SystemLogFileMaxNum setting in the <code>ssmlprocessor.conf</code> file.
Maximum size of the log file	SystemLogFileMaxSize setting in the <code>ssmlprocessor.conf</code> file.
Date/time format of the log file	SystemLogTime setting in the <code>ssmlprocessor.conf</code> file.

For more information, see the *Dialogic® Vision™ VoiceXML Administration Manual*.

Log file format

The format of each SSML Processor system log message is:

```
[timestamp] [severity] [category] [channel ID:thread ID] message
```

The following table describes the fields in an SSML Processor system log message:

Field	Description						
timestamp	Date and time of the log message, specified in local server time by default. Use the SystemLogTime setting in the <i>ssmlprocessor.conf</i> file to change the time format to Greenwich Mean Time (GMT). This also changes the time format in the log file name. For information, see the <i>Dialogic® Vision™ VoiceXML Administration Manual</i> .						
severity	Level of the log message. Valid values (in order of decreasing severity and increasing verbosity): - FATAL (a severe malfunction from which the VoiceXML Interpreter process cannot recover) - ERROR - WARNING - INFO1 (default) - INFO2 - INFO3 - INFO4 - INFO5 (highest/most verbose level of detail)						
category	General category to which the message refers: <table border="1"> <thead> <tr> <th>Label</th><th>Description</th></tr> </thead> <tbody> <tr> <td>TTS Engine</td><td>Messages related to TTS engines and their management.</td></tr> <tr> <td>Framework</td><td>Messages related to the general SSML Processor framework.</td></tr> </tbody> </table>	Label	Description	TTS Engine	Messages related to TTS engines and their management.	Framework	Messages related to the general SSML Processor framework.
Label	Description						
TTS Engine	Messages related to TTS engines and their management.						
Framework	Messages related to the general SSML Processor framework.						
channel ID	ID of the SSML Processor channel that generated the message, if the message came from an audio channel. Otherwise, the channel ID is replaced with a dash.						
thread ID	ID of the thread that generated the message. This can be used to track the progress of a single SSML request in the log file when there are several requests being processed simultaneously.						
message	Log message.						

VoiceXML application logs

VoiceXML applications running on the VoiceXML Interpreter create log messages that VoiceXML application developers can use to debug phone calls.

By default, application logging is set up as follows:

- Application logging is enabled.
- A separate log file is created for each call.
- The application log files are stored in the `vx/vxmlinterpreter/logs/subdir` directory, where **subdir** is the log subdirectory specified in the application configuration. For information about configuring applications, see *Managing VoiceXML applications* on page 61.

The following table describes how to change the defaults for application logging:

To...	Take this action...	For more information, see...
Disable application logging for all applications	Set the <code>VXMLAppLogsEnabled</code> setting in the <code>vxmlinterpreter.conf</code> file to false.	<i>Dialogic® Vision™ VoiceXML Administration Manual</i>
Change the base location for all log files	Modify the <code>VXMLAppLogBaseDir</code> setting in the <code>vxmlinterpreter.conf</code> file.	<i>Dialogic® Vision™ VoiceXML Administration Manual</i>
Change the subdirectory location for a specific application's log file	Modify the setting of the Log subdirectory field on the VoiceXML application configuration page in the Vision™ Console.	<i>Managing VoiceXML applications</i> on page 61

Note: To facilitate debugging, you can also send VoiceXML application messages to the VoiceXML Interpreter system log file. To place application log messages in the system log file, set the value of `SystemLogLevel` in the `vxmlinterpreter.conf` file to 3 or higher.

Log file format

The format for the VoiceXML Interpreter application log file name is:

```
[VXMLAppLogBaseDir]/{subdir}/yyyy/mm/hh/[Call ID].log
```

where:

- **subdir** is the log subdirectory specified in the application configuration. For more information, see *Managing VoiceXML applications* on page 61.
- **Call ID** is the unique identifier for the current call.

Log file format

The format of each VoiceXML Interpreter system log message is:

```
[timestamp] [severity] [category] [channel ID:thread ID] message
```

The following table describes the fields in a VoiceXML Interpreter system log message:

Field	Description																				
timestamp	Date and time of the log message, specified in local server time by default. Use the SystemLogTime setting in the <i>vxmliinterpreter.conf</i> file to change the time format to Greenwich Mean Time (GMT). This also changes the time format in the log file name. For information, see the <i>Dialogic® Vision™ VoiceXML Administration Manual</i> .																				
severity	Level of the log message. Valid values (in order of decreasing severity and increasing verbosity): - FATAL (a severe malfunction from which the VoiceXML Interpreter process cannot recover) - ERROR - WARNING - INFO1 (default) - INFO2 - INFO3 - INFO4 - INFO5 (highest/most verbose level of detail)																				
category	General category to which the message refers: <table border="1"> <thead> <tr> <th>Label</th><th>Description</th></tr> </thead> <tbody> <tr> <td>Xml Parser</td><td>XML parser related message.</td></tr> <tr> <td>VoiceXmlInterpreter</td><td>VoiceXML Interpreter flow/form interpretation algorithm related message.</td></tr> <tr> <td>I/O</td><td>Recognition/speech synthesis related message.</td></tr> <tr> <td>Debug</td><td>Debug message.</td></tr> <tr> <td>Telephony</td><td>Telephony related message.</td></tr> <tr> <td>Database</td><td>Database related message.</td></tr> <tr> <td>JS</td><td>ECMA/JavaScript related message.</td></tr> <tr> <td>Billing</td><td>Billing message.</td></tr> <tr> <td>Grammar</td><td>Grammar related message.</td></tr> </tbody> </table>	Label	Description	Xml Parser	XML parser related message.	VoiceXmlInterpreter	VoiceXML Interpreter flow/form interpretation algorithm related message.	I/O	Recognition/speech synthesis related message.	Debug	Debug message.	Telephony	Telephony related message.	Database	Database related message.	JS	ECMA/JavaScript related message.	Billing	Billing message.	Grammar	Grammar related message.
Label	Description																				
Xml Parser	XML parser related message.																				
VoiceXmlInterpreter	VoiceXML Interpreter flow/form interpretation algorithm related message.																				
I/O	Recognition/speech synthesis related message.																				
Debug	Debug message.																				
Telephony	Telephony related message.																				
Database	Database related message.																				
JS	ECMA/JavaScript related message.																				
Billing	Billing message.																				
Grammar	Grammar related message.																				
channel ID	ID of the VoiceXML Interpreter telephony channel that generated the message. Otherwise, the channel ID is replaced with a dash, such as for process-wide log messages.																				
thread ID	ID of the thread that generated the message. This can be used to track the progress of a single session in the log file when there are several requests being processed simultaneously.																				
message	Log message.																				

The format of these messages is:

```
timestamp [category] [severity] message
```

The following table describes the fields in a VoiceXML Interpreter application log message:

Field	Description														
timestamp	Date and time of the log message specified in UTC time as: <i>mm/dd/yy hh:mm:ss</i>														
category	Application category to which the message refers. The category types are: <table> <tr> <th>Label</th><th>Description</th></tr> <tr> <td>Recognition</td><td>Voice recognition message.</td></tr> <tr> <td>Events</td><td>VoiceXML event related message.</td></tr> <tr> <td>Variables</td><td>ECMA Script related message.</td></tr> <tr> <td>HTTP</td><td>HTTP message.</td></tr> <tr> <td>Log</td><td>VoiceXML <log> element message.</td></tr> <tr> <td>Flow</td><td>VoiceXML Interpreter flow/form Interpretation algorithm related message.</td></tr> </table>	Label	Description	Recognition	Voice recognition message.	Events	VoiceXML event related message.	Variables	ECMA Script related message.	HTTP	HTTP message.	Log	VoiceXML <log> element message.	Flow	VoiceXML Interpreter flow/form Interpretation algorithm related message.
Label	Description														
Recognition	Voice recognition message.														
Events	VoiceXML event related message.														
Variables	ECMA Script related message.														
HTTP	HTTP message.														
Log	VoiceXML <log> element message.														
Flow	VoiceXML Interpreter flow/form Interpretation algorithm related message.														
severity	Level of severity associated with the log message. Valid values: • ERROR • WARNING • INFO														
message	Log message.														

The following example shows a VoiceXML Interpreter application log message:

```
08/28/08 14:52:23 [Flow] [Info]
Playing prompt http://localhost:9002/vxml/examples/helloworld/helloworld.wav
```

SSML application logs

Interspersed with the SSML Processor system log messages are higher-level application log messages that are used to debug speech applications. When the SSML Processor is used as the audio source for the VX Integrated Media Platform, these messages are sent using HTTP POST using the VoiceXML Interpreter process that generated the SSML request. The messages are inserted into the application log file when the request has been processed.

The format of these messages is:

```
[timestamp] [category] [severity] message
```

The following table describes the fields in an SSML Processor application log message:

Field	Description						
timestamp	Date and time of the log message specified in UTC time as: mm/dd/yy hh:mm:ss.mmm						
category	Type of message: <table><tr><th>Label</th><th>Description</th></tr><tr><td>HTTP</td><td>Information on HTTP requests, responses, and caching.</td></tr><tr><td>Flow</td><td>Information on the general SSML Processor flow.</td></tr></table>	Label	Description	HTTP	Information on HTTP requests, responses, and caching.	Flow	Information on the general SSML Processor flow.
Label	Description						
HTTP	Information on HTTP requests, responses, and caching.						
Flow	Information on the general SSML Processor flow.						
severity	Level of severity associated with the message. Valid values: • ERROR • WARNING • INFO						
message	Log message.						

Logging and performance

When the VX Integrated Media Platform actively uses large numbers of channels, a large amount of logging can be produced. Due to system limits, such as the input/output rate of the machine operating system when writing to the hard drive, this might cause performance bottlenecks.

Set the system log level appropriately for the VoiceXML Interpreter, Call Server, SSML Processor, and Media Resource Function logs. For example, use log level 1 or lower for a deployment environment. In general, if the VX Integrated Media Platform uses a higher number of concurrent channels, change the log levels to a lower level of verbosity.

Use SystemLogFileMaxSize and SystemLogFileMaxNum at their default settings to actively prune log files.

8

CallPlacer interface

Overview of the CallPlacer interface

The CallPlacer interface is an XML-over-HTTP mechanism for initiating outbound sessions or calls. To initiate a session, an XML request is sent using an HTTP GET or POST (application/x-www-form-urlencoded) to the VoiceXML Interpreter URI `http://ip:port/callplacer` with ***cpxml***=***xml***.

where:

- ***ip*** is the IP address of the interpreter's web server.
- ***port*** is the web server's port.
- ***cpxml*** is the parameter name used in the application/x-www-form-urlencoded message body.
- ***xml*** is the CallPlacer XML described in this section.

The CallPlacer interface returns an XML document with an immediate response indicating that the VoiceXML Interpreter is dialing or an error has occurred. Periodically, a callback (XML-over-HTTP request) updating the status of the call as it proceeds can be made to the URI passed to the original request. This is optional.

This section describes the XML used to:

- Request a new session
- Respond to a request
- Perform a callback

This section also describes *CallPlacer XML document type definition (DTD)* on page 96.

Request XML

The following table shows the XML elements used to initiate a new session:

Element	Description
obnumber	Number to dial using the tel: URI syntax (RFC 2806). A tel: URI is converted to a SIP URI with the parameter user=phone, which indicates that the call should be placed over the PSTN network.
initialuri	Initial VoiceXML URI. If the value of INITIALPAGE is supplied, the first VoiceXML page is the one specified in the VoiceXML Interpreter's defaultinitialURI configuration setting. For more information, see the <i>Dialogic® Vision™ VoiceXML Administration Manual</i> .
servicenumber	ANI of the session or call.
callbackuri	(Optional) As the call progresses, callbacks can be sent periodically to this URI using an HTTP POST (application/x-www-form-urlencoded) operation with cpxml =XML. For a description of the XML used in the response, see <i>Response XML</i> on page 93.

For example:

```
<?xml version="1.0"?>
<placecall>
  <dial id="1">
    <obnumber>
      tel:+35312091919
    </obnumber>
    <initialuri>
      http://mydomain/myapp.vxml
    </initialuri>
    <servicenumber>
      43800800
    </servicenumber>
    <callbackuri>
      http://mydomain/callback.cgi
    </callbackuri>
  </dial>
</placecall>
```

Note: Multiple dial requests can be supplied in the placecall element.

Response XML

The following table contains the valid XML immediate responses to a placecall request:

Response type	Description
DIALING	VoiceXML Interpreter is dialing. A globally unique ID for the call is given as a child of the callid element. Syntax is: <pre><placecall> <response id="1" type="DIALING"> <callid> n032e-98aa-2348-98sdf82 </callid> </response> </placecall></pre>
UNAUTHORIZED	Call is not authorized. Syntax is: <pre><placecall> <response id="1" type="UNAUTHORIZED"/> </placecall></pre>
NORESOURCE	There are no free resources to place the call. Syntax is: <pre><placecall> <response id="1" type="NORESOURCE"/> </placecall></pre>
MAXEXCEEDED	Call is not authorized. Too many calls are in progress. Syntax is: <pre><placecall> <response id="1" type="MAXEXCEEDED"/> </placecall></pre>

Callback XML

When the DIALING response is received, the types of callbacks described in the following table can occur using an HTTP POST (application/x-www-form-urlencoded) operation with ***cpxml=XML***. The Call Placer interface expects an HTTP 200 header with an empty body as a response to these callbacks.

Response type	Description
ANSWERED	Session successfully started and the call was answered. Syntax is: <pre><placecall> <callback type="ANSWERED"> <callid> n032e-98aa-2348-98sdf82 </callid> </callback> </placecall></pre>
INCALL	Called repeatedly every 15 seconds to indicate that calls are in progress. If multiple calls are active with the same <callbackuri>, more than one <callid> is present. Syntax is: <pre><placecall> <callback type=" INCALL"> <callid> n032e-98aa-2348-98sdf82 </callid> <callid> n032e-98aa-2348-98sdf83 </callid> </callback> </placecall></pre>
CALLENDED	Session or call has ended. The <status> element returns the status of the call. Currently only OK is defined. Syntax is: <pre><placecall> <callback type="CALLENDED"> <callid> n032e-98aa-2348-98sdf82 </callid> <status> OK </status> </callback> </placecall></pre>
BUSY	Session or call received a busy status. Syntax is: <pre><placecall> <callback type=" BUSY"> <callid> n032e-98aa-2348-98sdf82 </callid> </callback> </placecall></pre>
NOANSWER	Session or call did not start or answer. Syntax is: <pre><placecall> <callback type=" NOANSWER"></pre>

Response type	Description
	<pre><callid> n032e-98aa-2348-98sdf82 </callid> </callback> </placecall></pre>
BADDESTINATION	Specified destination number was invalid. For example, an unsupported URI format. Syntax is: <pre><placecall> <callback type="BADDESTINATION"> <callid> n032e-98aa-2348-98sdf82 </callid> </callback> </placecall></pre>
FAIL	Session initiation failed for an unspecified reason. Syntax is: <pre><placecall> <callback type="FAIL"> <callid> n032e-98aa-2348-98sdf82 </callid> </callback> </placecall></pre>

CallPlacer XML document type definition (DTD)

The following example shows the document type definition (DTD) for the CallPlacer XML:

```
<!-- vision_placecall-1.0.dtd
<!-- Copyright (c) 2008 Dialogic Corporation    -->
<!-- http://www.dialogic.com                -->

<!ELEMENT placecall (dial | response | callback)>

<!ELEMENT dial (obnumber,initialuri,servicenumber,callbackuri?)>
<!ATTLIST dial    id    #REQUIRED>
<!ELEMENT response (callid?)>
<!ATTLIST response type (DIALING|UNAUTHORIZED|NORESOURCE|MAXEXCEEDED)
    "DIALING" #REQUIRED>

<!ELEMENT callback (callid, status?)>
<!ATTLIST callback type (ANSWERED|INCALL|CALLENDED|BUSY|NOANSWER)
    "ANSWERED" #REQUIRED>

<!ELEMENT obnumber (#PCDATA)>
<!ELEMENT initialuri (#PCDATA)>
<!ELEMENT servicenumber (#PCDATA)>
<!ELEMENT callbackuri (#PCDATA)>
<!ELEMENT callid (#PCDATA)>
<!ELEMENT status (OK) "OK">
```

9

Authorization and Usage Indication interface

Overview of the Authorization and Usage Indication interface

The Authorization and Usage Indication interface is an XML-over-HTTP mechanism modeled on the ETSI TS 101 321 Open Settlement Protocol (OSP) for authorization and usage exchange.

An authorization request is made at the start of the session or call and before the session is started or the call is answered. The subsequent authorization response sent in the HTTP response can:

- Authorize the call
- Modify the DNIS of an authorized call
- Deny the call

Modifying the DNIS of an inbound call allows the platform operator to hide private number ranges by manually replacing the local number with the original assigned number. At the end of the session or call, the VoiceXML Interpreter dispatches a call detail record (CDR) that contains the following information about a call:

- Time the call started
- Time the call ended
- Length of the call
- Transferred call information

A usage response is sent in the HTTP response to a usage indication request. If there is a problem sending this response, the VX Integrated Media Platform writes call detail records (CDRs) to the local disk. This occurs when the billing server is temporarily unavailable. For information, see *Call detail records* on page 105.

Message format

The following example shows the general format for OSP messages:

```
<?xml version="1.0"?>
<Message messageId="0" random="1234">
  ...
</Message>
```

The Message element contains one of the following child elements, called components. With the exception of UsageIndication, the components only appear once.

- AuthorizationRequest
- AuthorizationResponse
- UsageIndication
- UsageConfirmation

The following table describes the attributes of the Message element:

Attribute	Description
messageId (componentId)	Unique ID assigned to the message element and to its immediate descendants.
random	Random decimal value.

HTTP operations

The VoiceXML Interpreter uses the HTTP POST (application/x-www-form-urlencoded) operation with the field name osprequest. The field value contains the XML data contained in the AuthorizationRequest, AuthorizationResponse, UsageIndication, or UsageConfirmation messages. For more information, see *Authorization exchange* on page 99 and *Usage exchange* on page 102.

Authorization exchange

The authorization exchange consists of the following types of messages:

- AuthorizationRequest
- AuthorizationResponse

AuthorizationRequest

The following example shows the structure of an AuthorizationRequest message:

```
<AuthorizationRequest componentId="1">
  <Timestamp>
    2003-11-14T12:34:00Z
  </Timestamp>
  <CallId>
    0001D747-318C9A10-B3DB0006-5B39B574
  </CallId>
  <SourceInfo type="e164">
    35312345678
  </SourceInfo>
  <DestinationInfo type="e164">
    35318765432
  </DestinationInfo>
  <vision:CallType>
    INBOUND
  </vision:CallType>
  <Service/>
  <MaximumDestinations>
    0
  </MaximumDestinations>
</AuthorizationRequest>
```

Note: The VX Integrated Media Platform strips extra white space from actual requests to improve efficiency.

The following table lists the XML elements used in an AuthorizationRequest message:

Element	Abbreviation	Description
Timestamp	ts	Time at which the authorization request was initiated. The format is in coordinated universal time (UTC): yyyy-mm-ddthh:mm:ss.sssz. Millisecond accuracy is used.
CallId	ci	Globally unique identifier for the call that corresponds to the VoiceXML session variable session.connection.callid.
SourceInfo	si	ANI of the session or call. Valid values: - e164 - Default. Number format starts with the international prefix; for example, 3531234567. - uri - Number is a SIP URI.
DestinationInfo	di	DNIS of the session or call. Valid values: - e164 - Default. Number format starts with the international prefix; for example, 3531234567. - uri - Number is a SIP URI.
vision:CallType	vct	Indicates the type of call. Valid values: - INBOUND - OUTBOUND - TRANSFER
Service	N/A	Used according to Open Settlement Protocol (OSP) implementation, but not filled with values.
MaximumDestinations	N/A	Used according to OSP implementation, but not filled with values.

AuthorizationResponse

The following example shows the structure of an AuthorizationResponse message:

```
<AuthorizationResponse componentId="1">
  <Timestamp>2003-11-14T12:34:00Z</Timestamp>
  <Status>
    <Code>200</Code>
    <Description>success</Description>
  </Status>
  <TransactionId>000000</TransactionId>
</AuthorizationResponse>
```

Note: The VX Integrated Media Platform strips extra white space from actual responses to improve efficiency.

The following table lists the XML elements used in an Authorization Response message:

Element	Abbreviation	Description
Timestamp	ts	Time at which the authorization response was initiated. The format is in coordinated universal time (UTC): Format: yyyy-mm-ddthh:mm:ss.sssz. Millisecond accuracy is used.

Element	Abbreviation	Description
Status	N/A	Valid values: • 200 - Call authorized • 403 - Call not authorized • 404 - DNIS unknown
vision:Masquerade	N/A	Optional. If present, the VoiceXML Interpreter replaces the DNIS with this number in the VoiceXML session variables and logs.
TransactionId	ti	Not used.
vision:FeaturePermission	N/A	Optional. Controls permission for a call to a particular VoiceXML Interpreter feature. Format: name value Valid values for name : • asr • tts • transfer Valid values for value : • true - Default. Allows the call to use the named feature. • false - Disallows the call from using the named feature.

Usage exchange

The usage exchange consists of the following types of messages:

- UsageIndication request
- UsageConfirmation

UsageIndication request message

The following example shows the UsageIndication request message format:

```
<UsageIndication componentId="1">
  <Timestamp>
    2003-11-14T12:34:45Z
  </Timestamp>
  <Role>
    source
  </Role>
  <TransactionId>
    000000
  </TransactionId>
  <CallId>
    0001D747-318C9A10-B3DB0006-5B39B574
  </CallId>
  <SourceInfo type="e164">
    35312345678
  </SourceInfo>
  <DestinationInfo type="e164">
    35318765432
  </DestinationInfo>
  <vision:CallType>
    INBOUND
  </vision:CallType>
  <UsageDetail>
    <Service/>
    <Amount>
      45
    </Amount>
    <Increment>
      1.000000
    </Increment>
    <Unit>
      s
    </Unit>
    <StartTime>
      2003-11-14T12:34:00.000Z
    </StartTime>
    <EndTime>
      2003-11-14T12:34:45.000Z
    </EndTime>
    <TerminationCause>
      <TCCode>
        1016
      </TCCode>
      <Description>
        Normal call clearing
      </Description>
    </TerminationCause>
  </UsageDetail>
</UsageIndication>
```

Note: Blind transfers do not appear as additional UsageIndication elements.

The following table lists the XML elements used in the UsageIndication message:

Element	Description
Timestamp	Time at which the authorization request was initiated. The format is in coordinated universal time (UTC): yyyy-mm-ddthh:mm:ss.sssz. Millisecond accuracy is used.
Role	Required for a usage exchange message under the Open Settlement Protocol (OSP), but has no value.
TransactionId	Required for a usage exchange message under the Open Settlement Protocol (OSP), but has no value.
CallId	Globally unique identifier for the call that corresponds to the VoiceXML session variable session.connection.callid.
SourceInfo	ANI of the session (call). Valid values: - e164 - Default. Number format starts with the international prefix; for example, 3531234567. - uri - Number is a SIP URI.
DestinationInfo	DNIS of the session (call). Valid values: - e164 - Default. Number format starts with the international prefix; for example, 3531234567. - uri - Number is a SIP URI.
vision:CallType	Type of call. Valid values: - INBOUND - OUTBOUND - TRANSFER
vision:ParentCallID	For transferred calls, a child element of the UsageIndication message (component) that refers to the CallID of the original call. The original call can be inbound or outbound. The first UsageIndication message refers to the original inbound or outbound call. Subsequent UsageIndication messages, if present, refer to transferred calls and contain the <vision:ParentCallID> element. Note: An additional UsageIndication is not created for a blind transfer.
vision:ServiceNumber	A VX Integrated Media Platform service number for outbound calls.
UsageDetail	A block that contains details about a call. If an external system does not require this information, use the BillingOmitUsageDetail configuration setting in the <i>vxmliinterpreter.conf</i> file to omit it. For more information, see the <i>Dialogic® Vision™ VoiceXML Administration Manual</i> .
Service	Required for a usage exchange message under the Open Settlement Protocol (OSP), but has no value.
Amount	Duration of the call in seconds, rounded to nearest increment, as specified by the <Increment> element.
Increment	Increment in seconds for which values are output for the <StartTime>, <EndTime>, and <Amount> elements in the UsageDetail block. The value is always set to 1.000000. For example, if a call started at 2003-11-14T12:34:00.550Z, the time would be output in the <StartTime> element as: <StartTime>2003-11-14T12:34:01.000Z</StartTime>

Element	Description
Unit	Time designator associated with the Amount and Increment elements in the UsageDetail block. The value is always set to s (seconds).
StartTime	Time the call started, specified in the same format as the Timestamp element.
EndTime	Time the call ended, specified in the same format as the Timestamp element.
TerminationCause	Termination reason for a call, which includes a code in the <TCCode> element and plain text in the <Description> element.
TCCode	Termination code for a call. Valid values: • 1016 - Normal call clearing • 0017 - Busy • 0018 - No answer • 0038 - Network out of order • 0041 - Telephony error (system error)

UsageConfirmation message

The following example shows the UsageConfirmation message format:

```
<UsageConfirmation componentId="1">
  <Timestamp>
    2003-11-14T12:34:45Z
  </Timestamp>
  <Status>
    <Code>
      201
    </Code>
    <Description>
      cdr created
    </Description>
  </Status>
</UsageConfirmation>
```

The following table lists the XML elements used in the UsageConfirmation message:

Element	Description
Timestamp	Time the usage indication message request was initiated. Format: yyyy-mm-ddthh:mm:ss.sssz . Millisecond accuracy is used.
Status Code	Valid values: 201 - Success 500 - Error occurred

If there is a problem sending a UsageConfirmation message, the VX Integrated Media Platform writes call detail records (CDRs) to the local disk. This occurs when the billing server is temporarily unavailable. For information, see Call detail records.

Call detail records

By default, the VX Integrated Media Platform writes call detail records (CDRs) to disk when any of the following actions occur:

- CCXML session ends
- Endpoint hangs up the phone
- The billing server is temporarily unavailable.

You can direct the VX Integrated Media Platform to always record CDRs to disk, even when the Usage Indication requests are successful. To do this, set the `BillingWriteCDRToDisk` setting in the `vxmlconfig.conf` file to true.

The VX Integrated Media Platform records CDRs into a single text file in a condensed, non-XML format. The CDR has a directory structure and name that uses the following format:

YYYY/DD/MM/<hour>.cdr

The CDR file rolls over on the hour, every hour using UTC time. For example, the CDR file called `2009/08/06/05.cdr` is the file recorded at 5 am on the 8th of June 2009.

When a Usage Indication request is not fulfilled, the OSP XML CDR will be converted to the CDR file entry format and appended to the single CDR file. If the CDR file does not exist, it is created.

You can use a tool to parse these CDR files and attempt an upload of the CDRs to the Billing server at a later time when it becomes available again.

CDR entry format

The VX Integrated Media Platform uses the following format for CDR entries, with multiple name/value pairs separated by commas:

```
<name>=<value>,<name>=<value>,...;
```

A CDR entry omits fields that are not present in order to aid with parsing and disk-space efficiency.

If a CDR entry has been made in the file as a result of a failed Usage Indication request, the @ character is used as a suffix. This differentiates this type of CDR entry from CDRs entered because the Vision™ was configured to record all CDRs.

The following example shows a complete CDR entry:

```
ci=B28584CD 3B5011D9 80990007 EB592A8A,ts=2004-11-22T00:03:12Z,tc=1016,
dn=Normal call clearing,si=0035312091912,se='e164',di=170363161,dt='e164',
vct=INBOUND,vcm=VIDEO,vsn=170363161,it=1.000000,r1=source,at=15,ti=000000,ut=s,
st=2004-11-22T00:02:56.353Z,et=2004-11-22T00:03:11.382Z;\n
```

The following table illustrates the mapping between CDR abbreviations and Open Settlement Protocol (OSP) XML elements:

Abbreviation	OSP XML element	Description
at	Amount	Call duration, when this value is multiplied by the value of the Increment element.
ci	CallId	Call identifier.
di	DestinationInfo	Destination information or DNIS in one of the following formats:

Abbreviation	OSP XML element	Description
		- Full ITU-T Recommendation E.164 telephone number (11 numeric digits) - URL (12 digits) For example: <code>di=1290</code>
dn	Description	Call description.
dt	DestinationType	Destination type. Defaults to e64 or to a uri.
et	EndTime	Service end time, in GMT format. For example: <code>et=2007-11-08T08:00:58.345Z</code>
it	Increment	Increment of service measurement.
rl	Role	Role of the system generating the message. Valid values: - source - destination - other
se	SourceInfoType	Source type. Defaults to e164 or can be a URI. For example: <code>se=e164</code>
si	SourceInfo	Source information or ANI.
st	StartTime	Service start time, in GMT format. For example: <code>st=2007-11-08T08:00:51.355Z</code>
tc	TerminationCauseCode	Termination code. Valid values: 1016: Normal call clearing 0017: Busy 0018: No answer 0038: Network out of order 0041: Telephony error (system error)
ti	TransactionId	Transaction identifier.
ts	Timestamp	Time at which the component was generated, in GMT format.
ut	Unit	Units in which usage is recorded. Valid values: - s: seconds - pkt: packets (data grams) - b: bytes - p: fax pages - call: calls
vcm	vision:CallMode	Call mode. Valid values: - Voice - Video

Abbreviation	OSP XML element	Description
vct	vision:CallType	Call type. Valid values: • Inbound • Outbound • transfer
vsn	vision:Servicenumber	Service number when creating outbound calls from the CallPlacer interface.
vpc	vision:ParentCallID	For transferred calls, a child element of the UsageIndication message (component) that refers to the CallID of the original call. The original call can be inbound or outbound. The first UsageIndication message refers to the original inbound or outbound call. Subsequent UsageIndication messages, if present, refer to transferred calls and contain the <vision:ParentCallID> element. Note: An additional UsageIndication is not created for a blind transfer.

10 SIP interface

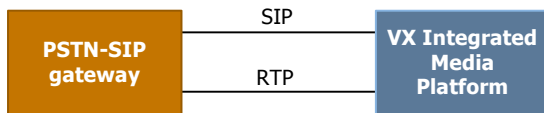
Overview of the SIP interface

The VX Integrated Media Platform integrates with SIP in two ways:

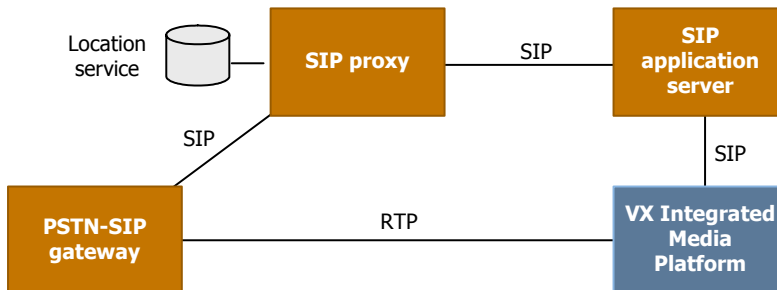
- Directly terminates IP calls with SIP signaling.
- Acts as a slave to application servers that establish RTP and VoiceXML sessions.

The following illustration shows how the VX Integrated Media Platform terminates IP calls:

- Directly terminates IP calls with SIP signaling.
- Acts as a slave to application servers that establish RTP and VoiceXML sessions.



The following illustration shows how the VX Integrated Media Platform integrates into a typical SIP network involving application servers, proxy servers, location servers, and servers:



The VX Integrated Media Platform SIP interface is described in more detail in the following topics:

- Interoperability
- Inbound calls
- Outbound calls
- VoiceXML exit information
- BYE reason header
- Call transfer

Interoperability

The VX Integrated Media Platform SIP interface works with systems that comply with the following specifications:

- RFC 3261 (SIP core)
- RFC 3263 (Locating SIP servers)
- RFC 3264 (SDP Offer/Answer)
- RFC 3515 (REFER method)
- RFC 2327 (SDP)
- RFC 1889 (RTP/RTCP)
- RFC 1890 (RTP profiles)
- RFC 2833 (RTP payload for DTMF digits)

The SIP interface supports the following audio/video codecs:

- AMR (RFC 3267)
- AMR IF2
- Comfort noise (payload type 13)
- G.711 A-law (payload type 8)
- G.711 mu-law (payload type 0)
- G723 (payload type 4)
- G279 (payload type 18)
- H.263 (payload type 34, RFC 2190)
- H.263+ (RFC 2429)
- H.264 (RFC 3984)
- MPEG-4 (RFC 3016)

Inbound calls

An inbound VoiceXML session is initiated through a SIP INVITE. The SDP Offer/Answer model (RFC 3264) is used for media negotiation. The INVITE usually contains the offer and the 200 OK response contains the answer SDP.

The following table lists the possible responses to the SIP INVITE:

Response	Description
200 OK	Success.
480 Temporarily Unavailable	Call not authorized.
503 Service Unavailable	No channels available to take call.

By default, the VX Integrated Media Platform handles incoming DTMF content in the following manner:

- If RFC 2833 is negotiated, then the server obtains the DTMF tone and duration through an RFC 2833 DTMF event.
- If RFC 2833 is not negotiated, then the server obtains the DTMF tone and duration from the body of the SIP INFO message.

If the joined connection is a SIP connection, then audio transcoding automatically occurs between the two endpoints, when required.

Variable mappings

The following table lists the SIP INVITE components and the corresponding VoiceXML session variables:

INVITE component	VoiceXML session variable
INVITE Protocol Name	session.connection.protocol.name (evaluates to sip)
INVITE Protocol Version	session.connection.protocol.version (evaluates to 2.0)
Request-URI	session.connection.protocol.sip.requesturi
Request-URI aai parameter	session.connection.aai (Also available through session.connection.protocol.sip.requesturi.aai)
Request-URI ccxml parameter	Stored as sub-properties of the session.connection.ccxml object (Also available through session.connection.protocol.sip.requesturi.ccxml)
Request-URI voicexml parameter	session.connection.protocol.sip.requesturi.voicexml (Also available through session.connection.initialuri) Note: If a query string is included in the parameter value, URL encode the special parameters. For example, replace the ? and = characters with %3F and %3D respectively.
Other parameters of the SIP Request-URI	Stored as sub-properties of the session.connection.protocol.sip.requesturi object
To header	session.connection.local.uri
From header	session.connection.remote.uri
Call-ID header	session.connection.callid
All headers of INVITE	Stored as sub-properties of the session.connection.protocol.sip.headers object

When set to true (the default), the media resource object (MIOSIP) configuration setting com.vision.miosip.dialog.FromToUseEntireSIPURI enables backward compatibility for VoiceXML applications that use the session.connection.remote.uri and session.connection.local.uri variables.

When MIOSIP receives a SIP BYE message, the Reason header (if present) is mapped to the _message variable associated with the VoiceXML connection.disconnect.hangup event.

Specifying the initial VoiceXML URI to invoke

To specify the initial VoiceXML URI to invoke, use the NETANN procedure. Include the VoiceXML URI to invoke as the **voicexml** parameter of the SIP Request-URI. If you omit the **voicexml** parameter, the VX Integrated Media Platform uses its application configuration system to map the user part of the To: field to an initial VoiceXML page.

For example:

```
sip:dialog@mediaserver.example.net; \
voicexml=http://vxmlserver.example.net/cgi-bin/script.vxml;aai=information
```

Session data update

You can obtain additional session protocol information after receiving a 200 OK response and answering a call; for example, to obtain the name of a videophone vendor after the underlying telephony connection is established.

To send additional information, use the Vision-OrigProtocolInfo header as part of the ACK to the 200 OK.

The syntax for the Vision-OrigProtocolInfo header is:

```
Vision-OrigProtocolInfo: name=sip,version=2,mode=video
```

This data maps to the `session.connection.protocol.sip.headers["Vision-Origprotocolinfo"]` session variable.

CCXML parameters

Since a CCXML application can initiate a session with the VX Integrated Media Platform, CCXML defines the following session variables within VoiceXML:

- `session.connection.ccxml.sessionid`
- `session.connection.ccxml.dialogid`
- `session.connection.ccxml.connectionid`
- `session.connection.ccxml.conferenceid`
- `session.connection.ccxml.namelist`
- `session.connection.ccxml.values`

These variables originate from parameters of the Request-URI without the `session.connection` prefix:

```
sip:dialog@10.0.0.1;ccxml.sessionid=1234;ccxml.dialogid=1234;ccxml.values.var1=value1
```

The `session.connection.ccxml.namelist` is not explicitly passed in the Request-URI. The VX Integrated Media Platform reconstructs the `namelist` from the `ccxml.values` parameters.

The `session.connection.ccxml` parameters are also available through the `session.connection.protocol.sip.requesturi["ccxml"]` session variable.

Proxy server

You can configure the VoiceXML Interpreter to use the proxy server specified in the `com.vision.miosip.rvsip.outboundProxyHost` configuration setting of the `vxmlinterpreter.conf` file. For more information, see the *Dialogic® Vision™ VoiceXML Administration Manual*.

Location server

The VoiceXML Interpreter can register with a location server, which is often co-located with the SIP proxy. To register the VoiceXML Interpreter with a location server, use the REGISTER method in conjunction with the configuration settings described in the following table:

Setting	Description
<code>com.vision.miosip.rvsip.locationServerAddress</code>	URI of the location server.
<code>com.vision.miosip.rvsip.userAgentAoR</code>	SIP address of record for the VX Integrated Media Platform
<code>com.vision.miosip.rvsip.userAgentContact</code>	Contact address.

Note: When the VoiceXML Interpreter is quiesced, it de-registers itself from the location server to allow for a graceful shutdown.

Load balancing and failover

Through its support of RFC 3263 behavior, the VX Integrated Media Platform provides resource discovery, load balancing, and transparent failover. The VX Integrated Media Platform uses the following configuration settings in the `vxmlinterpreter.conf` file to implement load balancing and failover behaviors:

- `com.vision.miosip.dialog.RemoteHosts`
- `com.vision.miosip.rvsip.outboundProxyHost`
- `com.vision.miosip.rvsip.locationServerAddress`

You can specify a comma-delimited list of IP addresses in the `com.vision.miosip.dialog.RemoteHosts` setting. The VoiceXML Interpreter performs a random robin algorithm on the list. If no response is received from an INVITE, the VoiceXML Interpreter transparently tries another address in the list. For more information, see the *Dialogic® Vision™ VoiceXML Administration Manual*.

For unreliable transport protocols such as UDP, the retransmission timer for an INVITE occurs with an exponential backoff of $2^n * T1$ and a maximum upper bound of $64 * T1$. Even with a recommended reduced setting for $T1$ (500ms) for high QoS networks, the initial INVITE takes 32 seconds to time out if the target server is not operational. TCP is preferable because the transport layer detects the error sooner and allows for faster failover. To use persistent TCP connections, set the `com.vision.miosip.rvsip.ePersistencyLevel` configuration setting appropriately.

Outbound calls

Trigger an outbound call by using a placecall request in the VoiceXML Interpreter CallPlacer interface. A placecall request triggers a SIP INVITE to one of the addresses specified by the com.vision.miosip.dialog.TelHosts configuration setting (if a tel: URI is used). The SDP Offer/Answer model (RFC 3264) is used for media negotiation. The INVITE contains the offer. The answer is expected in the 200 OK response.

For more information about the CallPlacer interface, see *Overview of the CallPlacer interface* on page 91.

Variable mappings

The following table describes the mappings between CallPlacer XML elements and SIP INVITE headers:

CallPlacer XML element	INVITE header
obnumber	User part of the To SIP URI. The rest of the URI is specified by the configuration setting com.vision.miosip.dialog.TelHosts. Can be specified either as a number or by using the tel: URI syntax.
servicenumber	User part of the From SIP URI and Vision-ServiceNumber. The rest of the URI is specified by the configuration setting com.vision.miosip.rvsip.userAgentAoR. Can be specified either as a number or by using the tel: URI syntax. A tel: URI is converted to a SIP URI, with the parameter user=phone used to indicate that the call must be placed over the PSTN network.

The following table lists the possible responses to the INVITE:

Response	Description
100 Trying	Provisional response.
180 Ringing	Provisional response - starts the connecttimeout timer.
200 OK	Success.
404 Not Found	Bad destination.
408 Request Timeout	No answer.
486 Busy Here	Engaged.
500 Server Internal Error	Fail.

The following table illustrates the mappings from the SIP response to the type attribute on the CallPlacer XML callback element:

SIP response	Value of type attribute
200 OK	ANSWERED
404 Not Found	BADDESTINATION
408 Request Timeout	NOANSWER
486 Busy Here	BUSY
500 Server Internal Error	FAIL
503 Service Unavailable	FAIL
603 Decline	NOANSWER

VoiceXML exit information

Variables from the VoiceXML <exit> element are returned in the SIP BYE body. To allow the application server to differentiate between a BYE resulting from a <disconnect> and one resulting from an <exit>, an application can use either parameter `_reason=disconnect` or `_reason=exit`. This is one way that a CCXML application can communicate with the VX Integrated Media Platform.

Note: An application can use either the `expr` attribute or the `namelist` attribute with the <exit> element, but not both.

There are two formats in which data can be returned to application server:

- Encoded according to SIP-VXML
- Formatted as XML

Returned data encoded as SIP-VXML

If the `ExitDataXML` configuration setting in the `vxmlinterpreter.conf` file is set to `false`, then returned data is encoded as specified in SIP-VXML.

Namelist variables are first converted to a string and encoded in the message body using the `application/x-www-form-urlencoded` format content type. The behavior resulting from specifying an ECMAScript object or a recording variable in the `namelist` is not defined.

The following examples show how data is returned encoded as SIP-VXML. For more information about SIP-VXML encoding, see *SIP Interface to VoiceXML Media Services*, Burke *et al.*, IETF Internet-Draft, draft-burke-vxml-2, November 2006.

Example 1

This example shows how the value of the `expr` attribute of VoiceXML <exit> is encoded and placed in the SIP BYE message:

VoiceXML:

```
<exit expr="'OK'"/>
```

Data in SIP BYE message:

```
_exit=OK&_reason=exit
```

Example 2

This example shows how the value of the `namelist` attribute of VoiceXML <exit> is encoded and placed in the SIP BYE message:

VoiceXML:

```
<exit namelist="myvar anothervar"/>
```

Data in SIP BYE message:

```
myvar=1&anothervar=2&__reason=exit
```

Returning data in XML format

If the `ExitDataXML` configuration setting in the `vx/vxmlinterpreter/conf` file is set to `true`, then data is returned in XML form. In this case, the MIME type for the body data is `application/vxml-data+xml`.

The following examples show how data is returned in XML format.

Example 1

In this example, the value attribute of <return> is set to the ECMA string value of the expr attribute of VoiceXML <exit>:

VoiceXML:

```
<exit expr="'OK'"/>
```

Data in SIP BYE message:

```
<vxml-data version="1.0" xmlns="http://www.dialogic.com/2004/vxml-data">
  <return value="OK"/>
</vxml-data>
```

Example 2

In this example, the format of the namelist content follows the XML serialization mechanism specified in Semantic Interpretation for Speech Recognition (SISR), with the top-level ECMAScript variable names serialized. The namelist content comes from the namelist attribute for <exit>.

VoiceXML:

```
<exit namelist="myvar myobject anothervar"/>
```

Data in SIP BYE message:

```
<vxml-data version="1.0" xmlns="http://www.dialogic.com/2004/vxml-data">
  <namelist>
    <myvar>1</myvar>
    <myobject>
      <x>1</x>
      <y>2</y>
    </myobject>
    <anothervar>2</anothervar>
  </namelist>
</vxml-data>
```

BYE reason header

When the VX Integrated Media Platform receives a BYE message, if the Reason header is present, it is mapped to the `_message` variable associated with the VoiceXML `connection.disconnect.hangup` event. This feature is used for communicating the termination reason, such as fax detection.

Call transfer

The VX Integrated Media Platform provides the following types of call transfers:

- Bridge
- Blind
- Consultation

Note: In this topic, the A leg refers to the initial call made to or from the VX Integrated Media Platform.

Bridge transfer

A bridge transfer occurs when the VX Integrated Media Platform initiates a transfer between two parties, and the caller returns to the VX Integrated Media Platform after the transfer ends (when the caller disconnects from the third party). With a bridge transfer:

- The VX Integrated Media Platform is aware of the outcome of the transfer.
- The original caller is not disconnected in the event of a connection error.

A bridge transfer appears as a new SIP INVITE from the VX Integrated Media Platform. The audio mixing occurs in the Media Resource Function component of the VoiceXML Interpreter. This component also performs DTMF hotword detection on the A leg.

The dest attribute has different meanings, depending on whether the call is being transferred to a SIP target or a PSTN target:

For this transfer target...	The dest attribute specifies the...
SIP	Destination SIP URI of the transfer target. Place this SIP call by using the com.vision.miosip.rvsip.outboundProxyHost configuration setting.
PSTN	User part of the To SIP URI. The remainder of the URI is constructed using the URI from the com.vision.miosip.dialog.TelHosts configuration setting. The dest attribute is specified using the tel: URI syntax and can include the custom parameters connecttimeout and maxtime. A tel: URI is converted to a SIP URI, with the user=phone parameter indicating that the call must be placed over the PSTN network.

The following example shows a tel: URI, along with its custom parameters:

```
tel:12345678;maxtime=60s&connecttimeout=10s.
```

The following table describes the custom parameters:

Custom parameter	Description
connecttimeout	Amount of time to wait for a final SIP response to the INVITE. When this time limit is reached, a CANCEL is issued and the transfer aborted. The timer for the connecttimeout starts when the 180 Ringing response is received.
maxtime	Maximum duration allowed for a call. When this time limit is reached, a BYE is issued to the outbound leg. The SIP INVITE contains the custom header Vision-ParentCallID, which equates to the Call-ID of the A leg.

While the outbound call is being attempted, the A leg hears the audio specified by the `com.vision.miosip.defaultTransferAudio` configuration setting.

The following table describes the mapping of SIP responses from the INVITE to the value of either the transfer form item variable or the VoiceXML event:

SIP response	Transfer form item variable / VoiceXML event
404 Not Found	<code>error.connection.baddestination</code>
408 Request Timeout	<code>noanswer</code>
486 Busy Here	<code>busy</code>
500 Server Internal Error	<code>unknown</code>
503 Service Unavailable	<code>noresource</code>
603 Decline	<code>noanswer</code>

The following table describes how the transfer form item variable is set when the outbound leg of a call is terminated:

Action	Transfer form item variable setting
If the called party hangs up (issues a BYE), the outbound call terminates.	<code>far_end_disconnect</code>
VoiceXML Interpreter terminates the outbound leg because the maximum call duration expired or because a hotword was detected.	<code>near_end_disconnect</code>

Blind transfer

A blind transfer occurs when the VX Integrated Media Platform initiates a transfer between two parties and then detaches from the call *before* the transfer takes place. With a blind transfer:

- The VX Integrated Media Platform is not aware of the outcome of the transfer.
- The original caller is disconnected if there is a transfer error.

A blind transfer uses the SIP REFER method (RFC 3515). After a successful response to the REFER message, the VX Integrated Media Platform generates a `connection.disconnect.transfer` and issues a BYE to drop the call.

The following table describes the attributes of the transfer element:

Attribute	Description
<code>dest</code>	Sets the user part of the SIP URI specified by the Refer-To header in the REFER message.
<code>aai</code>	Sets the aai part of the SIP URI specified by the Refer-To header in the REFER message.

The following example shows the use of a REFER message:

```
REFER sip:4708@10.3.1.52:5060 SIP/2.0
From: <sip:2078@dialogic.com>
To: <sip:10.3.1.52>;tag=ds-42d6-bb2c
Contact: sip:2078@10.3.2.89:5060
Call-ID: 12a1231bc323429@dialogic.com
CSeq: 1 REFER
Content-Length: 0
Refer-To: sip:1234568@voxbuilder.com;connecttimeout=2s;maxtime=60s
Via: SIP/2.0/UDP 10.0.0.99:5060
```

Valid responses to the REFER message are:

- 202 Accepted
- 500 Server Internal Error if an error occurs

Consultation transfer

A consultation transfer occurs when the VX Integrated Media Platform initiates a transfer between two parties and then detaches from the call after the transfer takes place. With a consultation transfer:

- The VX Integrated Media Platform is aware of the outcome of the transfer.
- The original caller is not disconnected in the event of a connection error.

A consultation transfer is also called a supervised transfer.

The following table describes the consultation transfer process:

Task	Description
1	The VX Integrated Media Platform creates a new call leg by issuing an INVITE message to the outbound server.
2	If the server accepts the call, the VX Integrated Media Platform issues a REFER message on the new call leg. The Refer-To header contains a Replaces field that holds the Call ID value for the new leg. This causes the new call leg to be replaced by the initial call leg on the remote UAC.
3	Control returns to the VoiceXML context.
4	A <connection.disconnect.transfer> event is issued.

11 Fine tuning the VX Integrated Media Platform configuration

Overview of fine tuning the configuration

After using the Vision™ Console to configure the VX Integrated Media Platform, you can fine tune the configuration by editing the configuration files. This topic describes how to fine tune the following types of configuration information:

- Media board
- Telecom (including signaling and media capabilities)
- H.100 clocking

For information about using the Vision™ Console to configure the VX Integrated Media Platform, see *Overview of configuring the VX Integrated Media Platform* on page 21.

Fine tuning the media board configuration

To fine tune the media board configuration for the VX Integrated Media Platform, modify the following media board configuration files:

- *oamsys.cfg*
- Media board configuration files

These files reside in the *vx/callserver/conf/cg* directory.

[oamsys.cfg file](#)

The *oamsys.cfg* file that ships with the VX Integrated Media Platform includes placeholders within <> brackets, such as **<IP-CG_0>**. Replace these placeholders with appropriate values. The values that do not have brackets are default values that you can replace, if necessary.

Specify the following settings in the *oamsys.cfg* file to configure the media boards:

Keyword	What to do
IPC.AddRoute[0].DestinationAddress	Replace <IP-CG_0> with the IPv4 address for the media board Ethernet interface.
IPC.AddRoute[0].Mask	Replace <SUBNET_MASK> with the subnet mask for the IPv4 address specified in IPC.AddRoute[x].DestinationAddress.
IPC.AddRoute[0].Interface	Specify 2 in place of 1 to associate Ethernet interface 2 with IPv4 connections. Leave the default 1 to associate Ethernet interface 1 with IPv4 connections.
IPC.AddRoute[1].serverAddress	Replace <server_IP> with the IPv4 address of the router.

If the platform is configured to use two or three media boards, specify the settings for the other media board. The setting names are the same, except that

<IP.CG_1> indicates the IPv4 address of the second media board and
<IP.CG_2> indicates the IPv4 address of the third media board.

The following example shows the settings in the *oamsys.cfg* file for a VX Integrated Media Platform with two media boards.

```
#-----
# oamsys.cfg - VX 5000
#-----

[cg6565-8_0]
Product                = CG_6565_8
Number                 = 0
Bus                    = 2
Slot                   = 3
IPC.AddRoute[0].DestinationAddress = <IP.CG_0>
IPC.AddRoute[0].Mask       = <SUBNET_MASK>
IPC.AddRoute[0].Interface  = 1
IPC.AddRoute[1].DestinationAddress = 0.0.0.0
IPC.AddRoute[1].Mask       = 0.0.0.0
IPC.AddRoute[1].serverAddress = <server_IP>

Clocking.HBus.ClockMode      = STANDALONE

File = /opt/nms/vx/callserver/conf/cg/board-0.cfg

[cg6565-8_1]
Product                = CG_6565_8
Number                 = 1
Bus                    = 2
Slot                   = 2
IPC.AddRoute[0].DestinationAddress = <IP.CG_1>
IPC.AddRoute[0].Mask       = <SUBNET_MASK>
IPC.AddRoute[0].Interface  = 1
IPC.AddRoute[1].DestinationAddress = 0.0.0.0
IPC.AddRoute[1].Mask       = 0.0.0.0
IPC.AddRoute[1].serverAddress = <server_IP>

Clocking.HBus.ClockMode      = STANDALONE

File = /opt/nms/vx/callserver/conf/cg/board-1.cfg
```

Media board configuration files

The media board configuration files (*board-n.cfg*) that ship with the VX Integrated Media Platform differ, depending on the server model, type of trunks (T1 or E1), and whether the board is an octal board.

To identify which board configuration file to edit, check the value of the File keyword in the *oamsys.cfg* file.

Note: The VX Integrated Media Platform can have up to three media boards. If it has multiple boards, you need to edit each media board configuration file. For example, if you have a two-board system, you must edit *board-0.cfg* and *board-1.cfg*.

Specifying T1 trunk settings

Specify the following settings in the media board configuration file for a media board that uses T1 trunks:

Keyword	Setting
NetworkInterface.T1E1[0..3].LineCode	For each trunk on the media board, optionally specify the ones density maintenance method used on the trunk line to maintain a clear channel transmission. Valid values: • B8ZS (Default) • AMI • AMI_ZCS • AMI_BELL • AMI_DDS • AMI_GTE
NetworkInterface.T1E1[0..3].FrameType	For each trunk on the media board, specify the T1 trunk framing format for the current trunk. Valid values: • D4 • ESF (Default)

The following example shows the default settings in the media board files for the ISDN audio and ISDN video models. Replace the bold default values, if necessary.

```
# T1 SPECIFICS
#
# If needed, edit the following parameters specific to your T1
# configuration.
NetworkInterface.T1E1[0..7].Type           = T1
NetworkInterface.T1E1[0..7].Impedance      = DSX1
NetworkInterface.T1E1[0..7].LineCode      = B8ZS
NetworkInterface.T1E1[0..7].FrameType    = ESF
NetworkInterface.T1E1[0..7].CRCMFMode      = C440N
NetworkInterface.T1E1[0..7].SignalingType  = PRI
NetworkInterface.T1E1[0..7].D_Channel      = ISDN
MaxChannels = 512
```

Specifying E1 trunk settings

Specify the following settings in the media board configuration file for a media board that uses E1 trunks:

Keyword	Setting
NetworkInterface.T1E1[0..3].Impedance	For each trunk on the media board, specify the type of cable connecting the board to the telephone network. Valid values: - G703_75_OHM - (Default) For G.703-standard 75 ohm cables. - G703_120_OHM - For G.703-standard 120 ohm cables. Note: For certain signal entry panels, set the media board trunks to 120 ohm, even if you use 75 ohm lines. For more information, see the <i>CG 6565 Installation and Developer's Manual</i>.
NetworkInterface.T1E1[0..3].LineCode	For each trunk on the media board, optionally specify the ones density maintenance method used on the trunk line to maintain a clear channel transmission. Valid values: - AMI - HDB3 (Default)

The following example shows the settings in the board file for the ISDN audio and ISDN video models. Replace the bold default values, if necessary:

```
# E1 SPECIFICS
#
# If needed, edit the following parameters specific to your E1
# configuration.
NetworkInterface.T1E1[0..7].Type = E1
NetworkInterface.T1E1[0..7].Impedance           = G703_120_OHM
NetworkInterface.T1E1[0..7].LineCode           = HDB3
NetworkInterface.T1E1[0..7].FrameType            = CEPT
NetworkInterface.T1E1[0..7].CRCMFMode            = C440N
NetworkInterface.T1E1[0..7].SignalingType        = PRI
NetworkInterface.T1E1[0..7].D_Channel            = ISDN
MaxChannels = 512
```

Fine tuning the telecom configuration

Note: This topic applies to all VX Integrated Media Platform models except IP-only audio and IP-only video.

To fine tune the telecom configuration for the VX Integrated Media Platform, modify the *telecom.conf* file. This file resides in the */vx/callserver/conf/* directory. The *telecom.conf* file that ships with the VX Integrated Media Platform includes placeholders within *<>* brackets, such as *<ISDN_TYPE>* (for ISDN models). You must replace these placeholders with appropriate values.

This topic contains the following subsections:

- Settings for all models except IP-only audio and IP-only video
- Setting for ISDN models
- Settings for ISUP models that use Vision™ Signaling Servers
- Settings for ISUP models that use native ISUP signaling

Settings for all models except IP-only audio and IP-only video

Specify the following parameter types for all VX Integrated Media Platform models except IP-only audio and IP-only video:

Parameter type	What to specify
Trunk	Trunks that the media board uses. Comment out the other trunks. For more information, see <i>Trunk settings</i> on page 127.
Route	Information for each route. For more information, see <i>Route settings</i> on page 129.
DTMF	Determines whether the VX Integrated Media Platform can: • Accept an incoming SIP INFO message with DTMF content. • Detect in-band DTMF in 3G-324M communications. • Send an outbound DTMF. You can also specify the length of DTMF events that the server generates. For more information, see <i>DTMF settings</i> on page 129.

Trunk settings

Use the Trunk keyword to specify the trunks that the media board uses. Comment out the other trunks in the file. The following table describes the parameters for the Trunk keyword that you can change if the defaults do not describe your configuration:

Parameter	Description
<i>trunk_num</i>	Unique logical trunk identifier. Valid values: 1 - 128.
<i>sig_board_num</i>	Logical signaling board identifier, referring to a board previously declared with the ISDN_board or ISUP_board keywords. Valid values: 1 - 8.

Parameter	Description
<i>media_board_num</i>	Logical board index of the media board containing the trunk, referring to a board previously declared with the Media_board keyword. Valid values: 1 - 8.
<i>trunk_vendor_id</i>	Vendor identifier of the trunk. Valid values: 0 - 15
<i>trunk_type</i>	(Optional) Type of trunk. Valid values: • E1 (30 media channels) • T1 (23 media channels) • E1RAW (31 media channels) • T1RAW (24 media channels) Default: E1.
<i>isdn_equipment</i>	(ISDN models only) Type of equipment connected to the trunk. Valid values: • NT = Network equipment • TE = Terminal equipment Default: TE
<i>-r channel_reserved</i>	(ISUP models only) Reserved channel. Use the -r switch to declare a voice channel as unusable for audio purposes. The channel-reserved value corresponds to the voice timeslot to reserve. Channels are numbered from 0 to (maximum number of voice timeslots) - 1. Valid values: • T1 trunks: 0 - 23 • E1 trunks: 0 - 30

Route settings

Use the Route keyword to specify information for each route. The following table describes the parameters for the Route keyword that you can change if the defaults do not describe your configuration:

Parameter	Description
route_num	Route identifier. Valid values: 1 - 16
trunk_num	One or more logical trunk identifiers for each trunk in the route. The trunk identifiers refer to trunks previously declared with the Trunk keyword. Valid values: 0 - 15
select_strategy	(Optional) Defines the route circuits reserved for outgoing calls, and specifies how the circuits are selected. Valid values: - FROM_TOP (default) - FROM_BOTTOM - DESCENDING - ASCENDING For more information, see <i>Values for the PSTN routes group</i> on page 42.

The following *telecom.conf* file example shows how to configure two routes on four ISDN trunks. One route contains trunk 2 with circuit selection strategy DESCENDING. The other route contains the three remaining trunks with the default circuit selection strategy (FROM_TOP).

```
# Four trunks (e.g. on a CG board with four EIs)
Trunk 1 1 3 0 E1 TE
Trunk 2 1 3 1 E1 TE
Trunk 3 1 3 2 E1 TE
Trunk 4 1 3 3 E1 TE
# Route 1 contains trunk 2 - selection strategy = DESCENDING
Route 1 2 DESCENDING
# Route 2 contains trunks 1, 3 and 4 - default selection strategy
Route 2 1 3 4
```

For more information about configuring the Call Server, see the *Dialogic® Vision™ Call Server Administration Manual*.

DTMF settings

The following table describes the DTMF parameters that you can change if the defaults do not correctly describe your configuration:

Parameter	Description
acceptInboundDTMFInSIPINFO	Indicates whether the VX Integrated Media Platform can accept an incoming SIP INFO message with DTMF content. Valid values: - TRUE: Server can accept an incoming SIP INFO message with DTMF content when RFC 2833 is not negotiated. - FALSE: Server cannot accept an incoming SIP INFO message with DTMF content. It rejects the message by issuing a 415 Media not supported response. Default: FALSE

Parameter	Description
detectInbandDtmfInVideoCalls	Indicates whether the VX Integrated Media Platform can detect in-band DTMF on 3G-324M mobile video calls. Valid values: 0: Platform does not detect in-band DTMF in mobile video calls. 1: Platform detects in-band DTMF when audio transcoding is required. 2: Platform always detects in-band DTMF. This forces audio transcoding, even if the audio codecs on both call legs are the same. Default: 1 This setting is useful for supporting handsets that do not support User Input Indication (UII). Note: Detecting in-band DTMF uses additional DS0 resources because each transcoded channel requires two DS0 resources.
dtmfEventDuration	Duration of an RFC 2833 DTMF event, in milliseconds. The VX Integrated Media Platform generates a DTMF event after it receives an H.245 User Input Indication message containing a DTMF from a 3G network. Default: 80 for audio setup; 300 for video setup
dtmfEventNotification	Indicates whether to notify the CCXML engine that a DTMF was received on a call leg. Valid values: - TRUE: Notify the CCXML engine. - FALSE: Do not notify the CCXML engine. Default: FALSE
outboundDTMFConfig	Determines how the VX Integrated Media Platform sends an outbound DTMF. Valid values: 0: Server always sends an outbound DTMF according to RFC 2833, if the other party supports RFC 2833. 1: Server always sends an outbound DTMF in a SIP INFO message. 2: Server sends an outbound DTMF according to RFC 2833, if the other party supports RFC 2833. If the other party does not support RFC 2833, the server sends the DTMF in a SIP INFO message. Default: 0

Setting for ISDN models

For the ISDN audio or ISDN video model, the default ISDN type for the ISDN_board keyword is ETSI (for ETSI ISDN). Replace this value, with any of the following values, if necessary:

- ATT_5E10 = AT&T 5
- AUSTEL_1 = Australian Telecom 1
- DMS100NT_DMS250 = Northern Telecom

- DMS250NTT = Nippon Telegraph Telephone
- ECMA_QSIG = ECMA 143
- ESS10ATT_4ESS = AT&T 4
- ESSKOREAN_OP = Korean operator
- NI2 = US National ISDN 2
- QSIGHK_TEL = Hong Kong Telecom
- TAIWAN_OP = Taiwanese operator
- VN6 = France Telecom
- VN6 NT_DMS = Northern Telecom

Settings for the ISUP models that use Vision™ Signaling Servers

For ISUP audio and video models that use Vision™ Signaling Servers, replace the parameters for the VS_isup_server keyword, as shown in the following table:

VS_isup_server 1 1 **<ISUP_START_CIRCUIT>** **<SS7_NODE_PC>**
<SS7_SWITCH_TYPE>

Parameter	Description
<ISUP_START_CIRCUIT>	Circuit identifier of the first circuit that the signaling server will handle.
<SS7_NODE_PC>	Point code assigned to the signaling server in the ISUP network.
<SS7_SWITCH_TYPE>	SS7 protocol variant that the signaling server uses. Valid values are: • 1ITUWHITE = ITU-T White Book • ITUBLUE = ITU-T Blue Book • ITU97 = ITU-T 1997 • Q767= ITU-T Q.767 • ETSIV2 = ETSI v2 (ETS 300 356) • ETSIV3 = ETSI v3 (ETS 300 356-1 – 1998) • JTTC = JTTC (Japan TTC) • JNTT = JNTT (Japan NTT) • ANSI88 = ANSI 88 (ref. T1.123 – 1988) • ANSI92 = ANSI 92 (ref. T1.113 – 1992) • ANSI95 = ANSI 95 (ref. T1.113 – 1995)

The following *telecom.conf* file example shows how to configure two Vision™ Signaling Servers. In this example, Signaling Server 1 services circuit 121 and higher, and uses point code 100. Signaling Server 2 services circuits 1 - 120, and uses point code 200. Both signaling servers support the ETSIV2 ISUP protocol variant.

```
# Signaling board
VS_isup_server 1 1 121 100 ETSIV2
VS_isup_server 2 2 1 200 ETSIV2
```

Settings for ISUP models that use native ISUP signaling

For ISUP audio and ISUP video models that use native ISUP signaling, replace the parameters for ISUP_board keyword, as shown in the following table:

ISUP_board 1 1 **<SS7_NODE_PC>** **<SS7_SWITCH_TYPE>**

Parameter	Description
<SS7_NODE_PC>	Point code of the gateway in the ISUP network. This value should be the same as the value for <SS7_NODE_PC> in the Overall MTP parameters section of the <i>MTP3cp1.cfg</i> file and the General configuration section of the <i>ISUPcp1.cfg</i> file.
<SS7_SWITCH_TYPE>	SS7 protocol variant. Declare the same SS7 protocol variant in the <i>isupcp1.cfg</i> file and in the <i>ss7load.bat</i> or <i>ss7load</i> file. Valid values: • 1ITUWHITE = ITU-T White Book • ITUBLUE = ITU-T Blue Book • ITU97 = ITU-T 1997 • Q767 = ITU-T Q.767 • ETSIV2 = ETSI v2 (ETS 300 356) • ETSIV3 = ETSI v3 (ETS 300 356-1 – 1998) • JTTC = JTTC (Japan TTC) • JNTT = JNTT (Japan NTT) • ANSI88 = ANSI 88 (ref. T1.123 – 1988) • ANSI92 = ANSI 92 (ref. T1.113 – 1992) • ANSI95 = ANSI 95 (ref. T1.113 – 1995)

Fine tuning the H.100 clocking configuration

Note: This topic applies to all VX Integrated Media Platform models except IP-only audio and IP-only video.

The VX Integrated Media Platform includes an H.100 clock manager that synchronizes the server's boards. By default, the H.100 clock manager starts automatically when the Call Server starts up, if the VX Integrated Media Platform has more than one media board.

Default H.100 clocking configuration

By default, H.100 clocking is configured for standalone operation (Clocking.HBus.ClockMode=STANDALONE), as specified in the *oamsys.cfg* file.

Clocking configuration for ISUP models

For ISUP models, the signaling timeslot is split on different trunks and boards for reliability. There is one signaling trunk defined on each media board. The following table describes the default clocking configuration for an ISUP system with one or two media boards:

Clock component	Default configuration
Primary master	Set on media board 1 and drives A_CLOCK. Uses the most reliable signaling trunk on the given media board as the first timing reference. Falls back to NETREF.
Secondary master	Set on media board 2 and drives B_CLOCK. Uses the primary master as the first timing reference. Falls back to the most reliable trunk on the given media board, other than the one driving the primary master. Note: For servers with one media board, there is no secondary master.
Signaling board	Slave on A_CLOCK. Falls back to B_CLOCK. Note: For servers with dedicated signaling links, the signaling board is not connected on the H.100 bus.

Clocking configuration for ISDN models

For ISDN models, all trunks carry a signaling link (D channel) for synchronization. The following table describes the default clocking configuration for an ISDN system with two media boards:

Clock component	Default configuration
Primary master	Set on media board 1 and drives A_CLOCK. Uses the most reliable signaling trunk on the given media board as the first timing reference. Falls back to NETREF.
Secondary master	Set on media board 2 and drives B_CLOCK. Uses the primary master as the first timing reference. Falls back to the most reliable trunk on the given media board, other than the one driving the primary master. Note: For servers with one media board, there is no secondary master.

H.100 clock manager configuration file

The H.100 clock manager uses the *priorities.xml* file to configure log settings, wait timeout intervals, timing references, and the board index of the clock source. This file resides in the *vx/clockmgr* directory.

The following example shows the default *priorities.xml* file for the ISUP models:

```
<?xml version="1.0" encoding="utf-8" ?>
<clockmanager>
  <log level="INFO" basefilename="clockmgr" outputdir="logs" filesize="1MB"
    maxfiles="5" />
  <txboard boardindex="1" />
  <clocksources>
    <source boardindex="0" trunk="1" priority="0" />
    <source boardindex="1" trunk="1" priority="0" />
  </clocksources>
</clockmanager>
```

The following table describes the elements and attributes in the *priorities.xml* file:

Element	Description
<clockmanager>	Root element of the <i>priorities.xml</i> configuration file.
<clocksources>	Root element for all timing references. This element is a child of the <clockmanager> element. All elements besides <clockmanager> are children of the <clocksources> element.

Element	Description												
<log>	Log settings for the H.100 clock manager. All attributes are optional. <table> <tr> <th>Attribute</th><th>Description</th></tr> <tr> <td>level</td><td> Log level for the H.100 clock manager log. Valid values: • CRITICAL • ERROR • WARNING • INFO Default: ERROR </td></tr> <tr> <td>basefilename</td><td> Base file name of the log output files. Default: <i>clkmgr</i> </td></tr> <tr> <td>outputdir</td><td> Output directory for the log files. Default: <i>vx/clockmgr/logs</i> </td></tr> <tr> <td>filesize</td><td> File size of each log file. Include a unit identifier (B, KB, or MB) with the value. If you do not include a unit identifier, the VX Integrated Media Platform uses MB. Valid values: 1B - <i>n</i>MB Default: 1MB Valid values: 1B - <i>n</i>MB Default: 1MB </td></tr> <tr> <td>maxfiles</td><td> Maximum number of files in the H.100 clock manager log. When all files are full, the logging subsystem overwrites the contents of the first file. Valid values: 1 - <i>n</i> Default: 5 </td></tr> </table>	Attribute	Description	level	Log level for the H.100 clock manager log. Valid values: • CRITICAL • ERROR • WARNING • INFO Default: ERROR	basefilename	Base file name of the log output files. Default: <i>clkmgr</i>	outputdir	Output directory for the log files. Default: <i>vx/clockmgr/logs</i>	filesize	File size of each log file. Include a unit identifier (B, KB, or MB) with the value. If you do not include a unit identifier, the VX Integrated Media Platform uses MB. Valid values: 1B - <i>n</i>MB Default: 1MB Valid values: 1B - <i>n</i>MB Default: 1MB	maxfiles	Maximum number of files in the H.100 clock manager log. When all files are full, the logging subsystem overwrites the contents of the first file. Valid values: 1 - <i>n</i> Default: 5
Attribute	Description												
level	Log level for the H.100 clock manager log. Valid values: • CRITICAL • ERROR • WARNING • INFO Default: ERROR												
basefilename	Base file name of the log output files. Default: <i>clkmgr</i>												
outputdir	Output directory for the log files. Default: <i>vx/clockmgr/logs</i>												
filesize	File size of each log file. Include a unit identifier (B, KB, or MB) with the value. If you do not include a unit identifier, the VX Integrated Media Platform uses MB. Valid values: 1B - <i>n</i>MB Default: 1MB Valid values: 1B - <i>n</i>MB Default: 1MB												
maxfiles	Maximum number of files in the H.100 clock manager log. When all files are full, the logging subsystem overwrites the contents of the first file. Valid values: 1 - <i>n</i> Default: 5												
<frequency>	Wait timeout for events. If defined, the clocking service executes in polling mode. If not defined, the clocking service executes in asynchronous mode (recommended). <table> <tr> <th>Attribute</th><th>Description</th></tr> <tr> <td>value</td><td> Time period, in milliseconds, for which the main loop waits for events before sending the timeout event. Valid values: -1 - <i>n</i> Default: -1 (CTA_WAIT_FOREVER) Required: No. </td></tr> </table>	Attribute	Description	value	Time period, in milliseconds, for which the main loop waits for events before sending the timeout event. Valid values: -1 - <i>n</i> Default: -1 (CTA_WAIT_FOREVER) Required: No.								
Attribute	Description												
value	Time period, in milliseconds, for which the main loop waits for events before sending the timeout event. Valid values: -1 - <i>n</i> Default: -1 (CTA_WAIT_FOREVER) Required: No.												

Element	Description								
<source>	Timing reference for H.100 bus clocking. All attributes are required. <table> <tr> <th>Attribute</th><th>Description</th></tr> <tr> <td>boardindex</td><td> OAM board index referenced by this clock source. Valid values: 0 - n Default: None. </td></tr> <tr> <td>trunk</td><td> Trunk number for the clock source. Valid values: 0 - n, where n is the total number of trunks supported by the board type. A value of 0 designates the board's internal oscillator (OSC) as the clock source. Default: None. </td></tr> <tr> <td>priority</td><td> Reliability of the trunk specified in the trunk attribute. Trunks with equivalent reliability can be given identical priority numbers. Valid values: 0 (best) - 99 (worst) Default: None. </td></tr> </table>	Attribute	Description	boardindex	OAM board index referenced by this clock source. Valid values: 0 - n Default: None.	trunk	Trunk number for the clock source. Valid values: 0 - n, where n is the total number of trunks supported by the board type. A value of 0 designates the board's internal oscillator (OSC) as the clock source. Default: None.	priority	Reliability of the trunk specified in the trunk attribute. Trunks with equivalent reliability can be given identical priority numbers. Valid values: 0 (best) - 99 (worst) Default: None.
Attribute	Description								
boardindex	OAM board index referenced by this clock source. Valid values: 0 - n Default: None.								
trunk	Trunk number for the clock source. Valid values: 0 - n, where n is the total number of trunks supported by the board type. A value of 0 designates the board's internal oscillator (OSC) as the clock source. Default: None.								
priority	Reliability of the trunk specified in the trunk attribute. Trunks with equivalent reliability can be given identical priority numbers. Valid values: 0 (best) - 99 (worst) Default: None.								
<txboard>	(ISUP models only) Defines the signaling board in the system. This element only applies when signaling is embedded in the voice trunks. <table> <tr> <th>Attribute</th><th>Description</th></tr> <tr> <td>boardindex</td><td> SS7 signaling board index defined by the board configuration. Default and only value: None. Required: Yes </td></tr> </table>	Attribute	Description	boardindex	SS7 signaling board index defined by the board configuration. Default and only value: None. Required: Yes				
Attribute	Description								
boardindex	SS7 signaling board index defined by the board configuration. Default and only value: None. Required: Yes								

Changing the default H.100 clocking configuration

To change the default H.100 clocking configuration, follow these steps:

Step	Action
1	Use the Vision™ Console to stop the H.100 Clock Manager service, as described in <i>Working with VX Integrated Media Platform services</i> on page 59.
2	Modify the <i>priorities.xml</i> file in the <i>vx/clockmgr</i> directory as appropriate. The elements and attributes for this file are described in <i>H.100 clock manager configuration file</i> on page 134.
3	Use the Vision™ Console to restart the H.100 Clock Manager service.

12 Glossary

A

ADTCP: An audio driver that provides a TCP interface to MIOSIP for rendering SSML fragments.

AMR: Adaptive multi-rate; an audio data compression scheme optimized for speech coding. This scheme was adopted by 3GPP and is used in video services.

ASR: Automatic speech recognition; ASR resources, called ASR engines in the MRCP framework, typically enable users of information systems to speak entries rather than punching numbers on a keypad. See also MRCP.

Authorization and Usage Indication interface: XML-over-HTTP mechanism that authorizes call sessions and gathers information for call detail reports.

B

blind transfer: A call transfer in which the originating caller is not announced and is connected directly to destination. In a blind transfer the Vision™ Server redirects the caller to the callee without remaining in the connection and does not monitor the outcome.

bridge transfer: A blind transfer in which the Vision™ Server redirects the caller to the callee and remains as a listener.

C

Call Server: Component of the Vision™ Server that manages call control and routing capabilities.

CallPlacer interface: XML-over-HTTP mechanism for initiating outbound sessions or calls for VoiceXML applications.

CCXML: Call Control Extensible Markup Language; a W3C Working Draft standard language for providing telephony call control support for dialog systems, gateways, and conferencing services.

CCXML application definition file: A file that maps individual CCXML applications to number ranges that trigger the execution of those applications.

clock: A periodic reference signal used for synchronization on a transmission facility, such as a telephony bus. See also clock master, clock slave, clock fallback.

clock master: A board that drives the clock signal for a system of boards connected by a bus cable. See also clock slave.

clock slave: A board that derives its clock signal from a bus cable; the clock signal is driven by the bus clock master. See also clock master.

consultation transfer: A call transfer in which the Vision™ Server initiates a transfer between two parties, but does not stay attached to the call once it is

successfully established. The caller remains connected to the Vision™ Server if the transfer fails.

D

DTMF: Dual tone multi frequency; an inband signaling system that uses two simultaneous voiceband tones for dialing. Also called touchtone. Some times DMTF is used to generally describe any telephony keypad press, even if tones are not generated.

G

G.711: An ITU PCM encoder/decoder specification for mu-law and A-law encoding.

H

H.100 bus: A TDM telephony bus standard for integrating hardware from various PC board vendors. The H.100 specification defines a ribbon cable bus that transports telephony voice data and signaling data across PCI boards. The H.100 bus is an interoperable superset of the H-MVIP and MVIP-90 telephony buses.

H.223: A protocol used to multiplex control and audio and video media on and off of a single DS0 within a trunk.

H.263: An ITU video compression standard. H.263 supports CIF, QCIF, SQCIF, 4CIF and 16CIF resolutions.

H.264: An ITU and ISO video compression standard that compresses video into lower bandwidth compared to H.263 and MPEG-4. H.264 is also called MPEG-4 Part 10.

I

INAP: Intelligent Network Application Part; an SS7 protocol that facilitates building platform-independent, transport-independent, and vendor-independent applications. Such applications include service switching points (SSPs), internet protocol (IP) applications, service control points (SCPs), enhanced services platforms, service circuit nodes, and other custom applications.

ISDN: Integrated services digital network; a standard for providing voice and data telephone service with all digital transmission and message-based signaling.

ISUP: ISDN user part; the SS7 protocol layer that allows for the establishment, supervision, and clearing of circuit-switched connections between two SS7 signaling points, such as central office switches. Despite its name, the ISUP layer is not unique to interconnecting. It is used to manage all types of circuit-switched connections.

ITU: International Telecommunications Union; an international standards body for telecommunications.

IVR: Interactive voice response; a telephony application in which callers interact with programs using recorded or synthesized voice prompts, DTMF digits, or speech recognition to query or deliver information.

M

Media Resource Function: Component of the Dialogic® Vision™ VX Integrated Media Platform that provides media processing including record, playback, and interfaces to speech recognition resources. The Media Resource Function is implemented by MIOSIP.

MIB: Management information base; an SNMP collection of objects that represent a managed node. Physically, a list of variables. Logically, a table with rows of variables.

MIOSIP: Implements the Media Resource Function of the Dialogic® Vision™ VX Integrated Media Platform. MIOSIP provides SIP call control, media processing over RTP, DTMF generation and recognition, and an MRCP client to automatic speech recognition (ASR) resources.

MPEG-4: An ISO/IEC standard for compressing multimedia data (video, audio, and speech).

MRCP: Media Resource Control Protocol; an application protocol for implementing automatic speech recognition (ASR) and text-to-speech services (TTS). MRCP provides a distributed system of ASR and TTS engines connected over an IP network.

MTP: Message transfer part; the SS7 protocol layers responsible for the reliable, in-sequence delivery of packets between two SS7 signaling points. The MTP functions include message routing, signaling link management, signaling route management, and congestion control.

MVIP-95: Device driver specification for H-MVIP, H.100, and H.110 telephony buses.

N

NETANN: Basic Network Media Services with SIP; an interface that enables applications in a SIP network to locate and invoke basic services on a media server. These services include network announcements, user interaction, and conferencing services. Also called RFC 4240.

O

OSP: Open Settlement Protocol; a European Telecommunications Standards Institute (ETSI) protocol used to exchange authorization, accounting, and usage information for IP telephony.

P

PSTN: Public switched telephone network; a public telephone network.

R

route: A connection path. On the PSTN network, a route is a logical collection of trunks. On the IP network, a route is a destination URL.

RTP: Real time transport protocol; a layer added to the internet protocol (IP) that addressed problems caused when real-time interactive exchanges (such as

audio data) are conducted over lines designed to carry packet-switched (connectionless) data.

S

- SCCP:** Signaling connection control part; an SS7 protocol that provides both connection-oriented and connectionless data transfer over an SS7 network. It extends the service provided by the SS7 MTP layers by adding extended addressing capabilities and multiple classes of service. The SCCP addressing capabilities allow a message to be addressed to an individual application or database within a signaling point. See also SS7.
- SDP:** Session description protocol, a protocol that defines a text-based format for describing streaming media sessions and multicast transmissions.
- Signaling Server:** An optional component of the Vision™ Server that provides redundant and scalable ISUP signaling.
- SIP:** Session initiation protocol. An IP signaling and telephony control protocol used mainly for voice over IP calls and multimedia communications. SIP relies on the session description protocol (SDP) for session description and the Real Time Transport Protocol (RTP) for actual transport.
- SRGS:** Speech Recognition Grammar Specification (SRGS); a syntax for representing the grammars used in speech recognition.
- SS7:** Signaling system 7; an out-of-band signaling system that provides fast call setup using circuit-switched connections and transaction capabilities for remote database interactions.
- SSML:** Speech Synthesis Markup Language; a proposed standard for enabling access to the internet using speech. SSML provides a standard way to control various aspects of speech (such as pronunciation, volume, pitch, and rate) over a variety of platforms.
- SSML Processor:** Component of the Dialogic® Vision™ VX Integrated Media Platform that processes SSML requests for audio and text-to-speech.

T

- T.38 fax:** A standard for real-time fax over IP that makes it possible for fax machines from different vendors to talk to each other over IP networks. The T.38 standard defines how to conduct group 3 facsimile transmission between terminals in which a portion of the transmission path between terminals includes (besides the PSTN or ISDN) an IP network such as the internet.
- TCAP:** Transaction capabilities application part; an SS7 protocol that provides applications with transaction support over the SS7 network. It enables the exchange of non-circuit related data, such as database queries and responses and remote feature invocation requests between SS7 signaling points. The TCAP layer relies on both the MTP and SCCP layers for message addressing and delivery.
- TDM:** Time division multiplexing; a technique for transmitting a number of separate data, voice, or video signals simultaneously over one communications medium by quickly interleaving a piece of each signal one after another.

telecom configuration file: File that provides information about the resources that interface with the Call Server and about other elements, such as the number of routes and the circuit selection.

trunk: The physical interface between the telephone network and the Vision™ Server. In telephone networks, a trunk is a shared connection between two switches. It differs from a line in that it is not dedicated to one subscriber or extension. T1 and E1 trunks carry 24 and 31 circuits, respectively.

TTS: Text-to-speech; a system that converts written language to speech.

V

Vision™ Console: Web-based configuration tool that configures the Vision™ Server.

VoiceXML: Voice Extensible Markup Language; a language that enables users to interact with the internet through voice recognition technology.

VoiceXML application configuration file: A file that maps individual VoiceXML applications to number ranges that trigger the execution of those applications.

VoiceXML Interpreter: Component of the Dialogic® Vision™ VX Integrated Media Platform that interprets VoiceXML dialogs.

VoiceXML Subsystem: Component of the Dialogic® Vision™ VX Integrated Media Platform that provides media processing for VoiceXML applications. The VoiceXML Subsystem consists of the VoiceXML Interpreter, SSML Processor, and Media Resource Function.

Index

A

- application 69
 - audio 70
 - CCXML 65
 - SSML 88
 - video 71
 - VoiceXML 61, 69, 85
- audio 70
- Authorization and Usage Indication interface 97
 - authorization exchange 99
 - message format 98
 - overview 97
 - usage exchange 102

B

- blind transfer 120
- board 123
- bridge transfer 119
- BYE message 118

C

- call detail records 105
- call processing 9
 - call transfer 119
 - inband-DTMF 9
 - inbound calls 111
 - outbound calls 91, 115
- Call Server 9
 - logging 76
 - starting, quiescing, and stopping 49
- callback XML 94
- CallPlacer interface 91
 - callback 94
 - outbound calls and SIP 115
 - overview 91

- request 92
- response 93
- callserver.conf file 76
- CCXML 11
 - applications 65
 - statistics 53
 - supported version 19
- CG 6565 board configuration files 124
- clocking 133
- conferencing 11
- configuration files 123
 - callserver.conf 76
 - media board configuration file 124
 - oamsys.cfg 123, 133
 - priorities.xml 134
 - ssmlprocessor.conf 82
 - telecom.conf 127
 - vxmllinterpreter.conf 79
- Configuration menu parameters 37
- configuring the Vision VX Integrated Media Platform 21
 - backing up a configuration 33
 - creating or revising a configuration 31
 - fine tuning the configuration 123
 - gathering information 22
 - restoring a configuration 34
- consultation transfer 121

D

- DHCP 28
- document type definition (DTD) 96
- documentation conventions 20
- DTMF 9, 111

E

events 9
exit element 116

F

fast call setup 12

G

gathering information 22

H

H.100 clocking 133
HTTP 69

I

inbound calls 111
interfaces 11
 Authorization and Usage Indication 97
 CallPlacer 91
 SIP 109
IP address 28
IP-only audio model 17, 70
IP-only video model 17, 71
ISDN audio model 12, 70
ISDN video model 13, 71
ISUP 19
ISUP audio model 14, 14, 70
ISUP models 14
ISUP redundant deployment model 16
ISUP scalable deployment model 15
ISUP video model 14, 14, 71

L

Linux 9
logging 75
 Call Server system logs 76
 logging and performance 89
 SSML application logs 88
 SSML Processor system logs 82
 VoiceXML application logs 85
 VoiceXML Interpreter system logs 79
logging into the platform 28, 35

M

media 69
media boards 123
Media Resource Function 9
models 9
 IP-only audio 17, 109
 IP-only video 17, 109
 ISDN audio 12
 ISDN video 13, 71
 ISUP audio 14, 14
 ISUP video 14, 14
Monitoring menu parameters 52
MRCP 19
MTP 19

N

NETANN 19

O

oamsys.cfg file 123
Operations menu parameters 49
OSP 19
outbound calls 91, 115

P

port capacity 46
Provisioning menu parameters 51

R

record utterance 73
related documentation 20
request XML 92
response XML 93
RFC 2833 19
RTSP 19

S

SIP interface 109
 BYE reason header 118
 call transfer 119
 inbound calls 111
 interoperability 110
 outbound calls 115

- VoiceXML exit information 116
- SISR 19
- SNMP 11
- software components 9
- SRGS 19
- SSML 19, 88
- SSML Processor 9
 - current activity 56
 - logging 82
 - starting and stopping 49
 - status 56
- ssmlprocessor.conf file 82
- standards 19
- statistics information 53
- status information 54, 55, 56
- T**
- telecom.conf file 127
- V**
- video 71
- Vision™ Console 21
 - accessing 30
 - backing up a configuration 33
- Configuration menu parameters 37
 - creating or revising a configuration 31
 - gathering information for 22
- Monitoring menu parameters 52
- Operations menu parameters 49
- Provisioning menu parameters 51
- restoring a configuration 34
- starting, quiescing, and stopping server services 59
- viewing platform information 60
- VoiceXML 69
 - application 61, 85
 - exit information 116
 - supported version 19
- VoiceXML Interpreter 9
 - current activity 54
 - logging 79
 - starting and stopping 49
 - status 55
- VoiceXML subsystem 9
- vxmlinterpreter.conf file 79